Multi-class Generalized Binary Search for Active Inverse Reinforcement Learning

Francisco S. Melo
INESC-ID/Instituto Superior Técnico
Portugal
fmelo@inesc-id.pt

Manuel Lopes
INRIA Bordeaux Sud-Ouest
France
manuel.lopes@inria.fr

Abstract

This paper addresses the problem of learning a task from demonstration. We adopt the framework of *inverse reinforcement learning*, where tasks are represented in the form of a reward function. Our contribution is a novel active learning algorithm that enables the learning agent to query the expert for more informative demonstrations, thus leading to more sample-efficient learning. For this novel algorithm (Generalized Binary Search for Inverse Reinforcement Learning, or GBS-IRL), we provide a theoretical bound on sample complexity and illustrate its applicability on several different tasks. To our knowledge, GBS-IRL is the first active IRL algorithm with provable sample complexity bounds. We also discuss our method in light of other existing methods in the literature and its general applicability in multi-class classification problems. Finally, motivated by recent work on learning from demonstration in robots, we also discuss how different forms of human feedback can be integrated in a transparent manner in our learning framework.

1 Introduction

Social learning, where an agent uses information provided by other individuals to polish or acquire anew skills, is likely to become one primary form of programming such complex intelligent systems (Schaal, 1999). Paralleling the social learning ability of human infants, an artificial system can retrieve a large amount of task related information by observing and/or interacting with other agents engaged in relevant activities. For example, the behavior of an expert can bias an agent's exploration of the environment, improve its knowledge of the world, or even lead it to reproduce parts of the observed behavior (Melo et al, 2007).

In this paper we are particularly interested in *learning from demonstration*. This particular form of social learning is commonly associated with *imitation* and *emulation* behaviors in nature (Lopes et al, 2009a). It is also possible to find numerous successful examples of robot systems that learn from demonstration (see the survey works of Argall et al, 2009; Lopes et al, 2010). In the simplest form of interaction, the demonstration may consist of examples of the right action to take in different situations.

In our approach to learning from demonstration we adopt the formalism of *inverse reinforcement learning* (IRL), where the task is represented as a *reward function* (Ng and Russel, 2000). From this representation, the agent can then construct its own policy and solve the target task. However, and unlike many systems that learn from demonstration, in this paper we propose to combine ideas from *active learning* (Settles, 2009) with IRL, in order to reduce the data requirements during learning. In fact, many agents able to learn from demonstration are designed to process batches of data, typically acquired before any actual learning takes place. Such data acquisition process fails to take advantage of any information the learner may acquire in early stages of learning to guide the acquisition of new data. Several recent works have proposed that a more *interactive* learning may actually lead to improved learning performance.

We adopt a Bayesian approach to IRL, following Ramachandran and Amir (2007), and allow the learning agent to actively select and query the expert for the desired behavior at the most informative situations. We contribute a theoretical analysis of our algorithm that provides a bound on the sample complexity of our learning approach and illustrate our method in several problems from the IRL literature.

Finally, even if learning from demonstration is the main focus of our paper and an important skill for intelligent agents interacting with human users, the abil-

ity to accommodate different forms of feedback is also useful. In fact, there are situations where the user may be unable to properly demonstrate the intended behavior and, instead, prefers to describe a task in terms of a *reward function*, as is customary in reinforcement learning (Sutton and Barto, 1998). As an example, suppose that the user wants the agent to learn how to navigate a complex maze. The user may experience difficulties in navigating the maze herself and may, instead, allow the agent to explore the maze and reward it for exiting the maze.

Additionally, recent studies on the behavior of naïve users when instructing agents (namely, robots) showed that the feedback provided by humans is often ambiguous and does not map in any obvious manner to either a reward function or a policy (Thomaz and Breazeal, 2008; Cakmak and Thomaz, 2010). For instance, it was observed that human users tend to provide learning agents with anticipatory or guidance rewards, a situation seldom considered in reinforcement learning (Thomaz and Breazeal, 2008). This study concludes that robust agents able to successfully learn from human users should be flexible to accommodate different forms of feedback from the user.

In order to address the issues above, we discuss how other forms of expert feedback (beyond policy information) may be integrated in a seamless manner in our IRL framework, so that the learner is able to recover efficiently the target task. In particular, we show how to combine both *policy and reward information* in our learning algorithm. Our approach thus provides a useful bridge between reinforcement learning (or learning by trial and error) and imitation learning (or learning from demonstration), a line of work seldom explored in the literature (see, however, the works of Knox and Stone, 2010, 2011, and discussion in Section 1.1).

The paper is organized as follows. In the remainder of this section, we provide an overview of related work on social learning, particularly on learning from demonstration. We also discuss relevant research in IRL and active learning, and discuss our contributions in light of existing work. Section 2 revisits core background concepts, introducing the notation used throughout the paper. Section 3 introduces our active IRL algorithm, GBS-IRL, and provides a theoretical analysis of its sample complexity. Section 4 illustrates the application of GBS-IRL in several problems of different complexity, providing an empirical comparison with other methods in the literature. Finally, Section 5 concludes the paper, discussing directions for future research.

1.1 Related Work

There is extensive literature reporting research on intelligent agents that learn from expert advice. Many examples feature robotic agents that learn simple tasks from different forms of human feedback. Examples include the robot Leonardo that is able to learn new tasks by observing changes induced in the world (as perceived by the robot) by a human demonstrating the target task Breazeal et al (2004). During learning, Leonardo provides additional feedback on its current understanding of the task that the human user can then use to provide additional information. We refer the survey works of Argall et al (2009); Lopes et al (2010) for a comprehensive discussion on learning from demonstration.

In this paper, as already mentioned, we adopt the inverse reinforcement learning (IRL) formalism introduced in the seminal paper by Ng and Russel (2000). One appealing aspect of the IRL approach to learning from demonstration is that the learner is not just "mimicking" the observed actions. Instead, the learner infers the purpose behind the observed behavior and sets such purpose as its goal. IRL also enables the learner to accommodate for differences between itself and the demonstrator (Lopes et al, 2009a).

The appealing features discussed above have led several researchers to address learning from demonstration from an IRL perspective. Abbeel and Ng (2004) explored inverse reinforcement learning in a context of *apprenticeship learning*, where the purpose of the learning agent is to replicate the behavior of the demonstrator, but is only able to observe a sequence of states experienced during task execution. The IRL formalism allows the learner to reason about which tasks could lead the demonstrator to visit the observed states and infer how to replicate the inferred behavior. Syed et al (Syed et al, 2008; Syed and Schapire, 2008) have further explored this line of reasoning from a game-theoretic perspective, and proposed algorithms to learn from demonstration with provable guarantees on the performance of the learner.

Ramachandran and Amir (2007) introduced *Bayesian inverse reinforcement learning* (BIRL), where the IRL problem is cast as a Bayesian inference problem. Given a prior distribution over possible target tasks, the algorithm uses the demonstration by the expert as evidence to compute the posterior distribution over tasks and identify the target task. Unfortunately, the Monte-Carlo Markov chain (MCMC) algorithm used to approximate the poste-

rior distribution is computationally expensive, as it requires extensive sampling of the space of possible rewards. To avoid such complexity, several posterior works have departed from the BIRL formulation and instead determine the task that maximizes the likelihood of the observed demonstration (Lopes et al, 2009b; Babes et al, 2011).

The aforementioned maximum likelihood approaches of Lopes et al (2009b) and Babes et al (2011) take advantage of the underlying IRL problem structure and derive simple gradient-based algorithms to determine the maximum likelihood task representation. Two closely related works are the *maximum entropy approach* of Ziebart et al (2008) and the *gradient IRL approach* of Neu and Szepesvari (2007). While the former selects the task representation that maximizes the likelihood of the observed expert behavior, under the maximum entropy distribution, the latter explores a gradient-based approach to IRL, but the where the task representation is selected so as to induce a behavior as similar as possible to the expert behavior.

Finally, Ross and Bagnell (2010) propose a learning algorithm that reduces imitation learning to a classification problem. The classifier prescribes the best action to take in each possible situation that the learner can encounter, and is successively improved by enriching the data-set used to train the classifier.

All above works are designed to learn from whatever data is available to them at learning time, data that is typically acquired before any actual learning takes place. Such data acquisition process fails to take advantage of the information that the learner acquires in early stages of learning to guide the acquisition of new, more informative data. *Active learning* aims to reduce the data requirements of learning algorithms by actively selecting potentially informative samples, in contrast with random sampling from a predefined distribution (Settles, 2009). In the case of learning from demonstration, active learning can be used to reduce the number of situations that the expert/human user is required to demonstrate. Instead, the learner should proactively ask the expert to demonstrate the desired behavior at the most informative situations.

Confidence-based autonomy (CBA), proposed by Chernova and Veloso (2009), also enables a robot to learn a task from a human user by building a mapping between situations that the robot has encountered and the adequate actions. This work already incorporates a mechanism that enables the learner to ask the expert for the right action when it encounters a situation in which it is less confident about the correct behavior. The system also allows the human user to provide corrective feedback as the robot executes the learned task.[1] The querying strategy in CBA can be classified both as *stream-based* and as *mellow* (see discussions in the survey works of Settles, 2009; Dasgupta, 2011). Stream-based, since the learner is presented with a stream of samples (in the case of CBA, samples correspond to possible situations) and only asks for the labels (*i.e.,* correct actions) of those samples it feels uncertain about. Mellow, since it does not seek highly informative samples, but queries any sample that is at all informative.

In the IRL literature, active learning was first explored in a preliminary version of this paper (Lopes et al, 2009b). In this early version, the learner actively queries the expert for the correct action in those states where it is most uncertain about the correct behavior. Unlike CBA, this active sampling approach is *aggressive* and uses *membership query synthesis*. Aggressive, since it actively selects highly informative samples. And, unlike CBA, it can select ("synthesize") queries from the whole input space. Judah et al (2011) propose a very similar approach, the *imitation query-by-committee* (IQBC) algorithm, which differs only from the previous active sampling approach in the fact that the learner is able to accommodate the notion of "bad states", *i.e.,* states to be avoided during task execution.

Cohn et al (2011) propose another closely related approach that, however, uses a different criterion to select which situations to query. EMG-AQS (Expected Myopic Gain Action Querying Strategy) queries the expert for the correct action in those states where the expected *gain of information* is potentially larger. Unfortunately, as discussed by Cohn et al (2011), the determination of the expected gain of information requires extensive computation, rendering EMG-AQS computationally costly. On a different line of work, Ross et al (2011); Judah et al (2012) address imitation learning using a no-regret framework, and propose algorithms for direct imitation learning with provable bounds on the regret. Finally, Melo and Lopes (2010) use active learning in a metric approach to learning from demonstration.

Our approach in this paper is a modified version of our original active sampling algorithm (Lopes et al, 2009b). We depart from the generalized binary search (GBS) algorithm of Nowak (2011) and adapt it to the IRL setting. To this purpose, we cast IRL as a (multi-class) classification problem and extend the GBS al-

[1] Related ideas are further explored in the *dogged learning* architecture of Grollman and Jenkins (2007).

gorithm of Nowak (2011) to this multi-class setting. We analyze the sample complexity of our GBS-IRL approach, thus providing the first active IRL algorithm with provable bounds on sample complexity. Also, to the extent of our knowledge, GBS-IRL is the first aggressive active learning algorithm for non-separable, multi-class data (Dasgupta, 2011).

We conclude this discussion of related work by pointing out that all above works describe systems that learn from human feedback. However, other forms of expert advice have also been explored in the agent learning literature. Price and Boutilier (1999, 2003) have explored how a learning agent can improve its performance by observing other similar agents, in what could be seen as "implicit" imitation learning. In these works, the demonstrator is, for all purposes, oblivious to the fact that its actions are being observed and learned from. Instead, the learned observes the behavior of the other agents and extracts information that may be useful for its own learning (for example, it may extract useful information about the world dynamics).

In a more general setting, Barto and Rosenstein (2004) discuss how different forms of supervisory information can be integrated in a reinforcement learning architecture to improve learning. Finally, Knox and Stone (2009, 2010) introduce the TAMER paradigm, that enables a reinforcement learning agent to use human feedback (in addition to its reinforcement signal) to guide its learning process.

1.2 Contributions

Our contributions can be summarized as follows:

- A *novel active IRL algorithm*, GBS-IRL, that extends generalized binary search to a multi-class setting in the context of IRL.
- The *sample-complexity analysis of GBS-IRL*. We establish, under suitable conditions, the exponential convergence of our active learning method, as a function of the number of samples. As pointed out earlier, to our knowledge ours is the first work providing sample complexity bounds on active IRL. Several experimental results confirm the good sample performance of our approach.
- A general discussion on how different forms of expert information (namely action and reward information) can be integrated in our IRL setting. We illustrate the applicability of our ideas in several simple scenarios and discuss the applicability of these different sources of information in face of our empirical results.

From a broader perspective, our analysis is a non-trivial extension of the results of Nowak (2011) to a multiclass setting, having applications not only on IRL but on any multiclass classification problem.

2 Background and Notation

This section introduces background material on Markov decision processes and the Bayesian inverse reinforcement learning formalism, upon which our contributions are developed.

2.1 Markov Decision Processes

A *Markov decision problem* (MDP) describes a sequential decision problem in which an agent must choose the sequence of actions that maximizes some reward-based optimization criterion. Formally, an MDP $\mathcal{M}$ is a tuple $\mathcal{M} = (\mathcal{X}, \mathcal{A}, \mathsf{P}, r, \gamma)$, where $\mathcal{X}$ represents the state-space, $\mathcal{A}$ the finite action space, P represents the transition probabilities, r is the reward function and γ is a positive discount factor. $\mathsf{P}(y \mid x, a)$ denotes the probability of transitioning from state x to state y when action a is taken, *i.e.,*

$$\mathsf{P}(y \mid x, a) = \mathbb{P}\left[X_{t+1} = y \mid X_t = x, A_t = a\right],$$

where each $X_t, t = 1, \ldots,$ is a random variable (r.v.) demoting the state of the process at time-step t and A_t is a r.v. denoting the action of the agent at time-step t.

A *policy* is a mapping $\pi : \mathcal{X} \times \mathcal{A} \to [0, 1]$, where $\pi(x, a)$ is the probability of choosing action $a \in \mathcal{A}$ in state $x \in \mathcal{X}$. Formally,

$$\pi(x, a) = \mathbb{P}\left[A_t = a \mid X_t = x\right].$$

It is possible to associate with any such policy π a *value-function*,

$$V^\pi(x) = \mathbb{E}_\pi\left[\sum_{t=0}^{\infty} \gamma^t r(X_t, A_t) \mid X_0 = x\right],$$

where the expectation is now taken over possible trajectories of $\{X_t\}$ induced by policy π. The purpose of the agent is then to select a policy π^* such that

$$V^{\pi^*}(x) \geq V^\pi(x),$$

for all $x \in \mathcal{X}$. Any such policy is an *optimal policy* for that MDP and the corresponding value function is denoted by V^*.

Given any policy π, the following recursion holds

$$V^\pi(x) = r_\pi(x) + \gamma \sum_{y \in \mathcal{X}} \mathsf{P}_\pi(x,y) V^\pi(y)$$

where $\mathsf{P}_\pi(x,y) = \sum_{a \in \mathcal{A}} \pi(x,a) \mathsf{P}_a(x,y)$ and $r_\pi(x) = \sum_{a \in \mathcal{A}} \pi(x,a) r(x,a)$. For the particular case of the optimal policy π^*, the above recursion becomes

$$V^*(x) = \max_{a \in \mathcal{A}} \left[r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathsf{P}_a(x,y) V^*(y) \right].$$

We also define the Q-function associated with a policy π as

$$Q^\pi(x,a) = r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathsf{P}_a(x,y) V^\pi(y)$$

which, in the case of the optimal policy, becomes

$$\begin{aligned} Q^*(x,a) &= r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathsf{P}_a(x,y) V^*(y) \\ &= r(x,a) + \gamma \sum_{y \in \mathcal{X}} \mathsf{P}_a(x,y) \max_{b \in \mathcal{A}} Q^*(y,b). \end{aligned} \tag{1}$$

2.2 Bayesian Inverse Reinforcement Learning

As seen above, an MDP describes a sequential decision making problem in which an agent must choose its actions so as to maximize the total discounted reward. In this sense, the reward function in an MDP encodes the *task* of the agent.

Inverse reinforcement learning (IRL) deals with the problem of recovering the task representation (*i.e.,* the reward function) given a demonstration of the behavior to be learned (*i.e.,* the desired policy). In this paper we adopt the formulation in Ramachandran and Amir (2007), where IRL is cast as a *Bayesian inference problem*, in which the agent is provided with samples of the desired policy, π^*, and it must identify the target reward function, r^*, from a general set of possible functions $\mathcal{R}$. Prior to the observation of any policy sample and given any measurable set $R \subset \mathcal{R}$, the initial belief that $r^* \in R$ is encoded in the form of a probability density function ρ defined on $\mathcal{R}$, *i.e.,*

$$\mathbb{P}[r^* \in R] = \int_R \rho(r) dr.$$

As discussed by Ramachandran and Amir (2007); Lopes et al (2009b), it is generally impractical to explicitly maintain and update ρ. Instead, as in the aforementioned works, we work with a finite (but potentially very large) sample of $\mathcal{R}$ obtained according to ρ. We denote this sample by $\mathcal{R}_\rho$, and associate with each element $r_k \in \mathcal{R}_\rho$ a *prior probability* $p_0(r_k)$ given by

$$p_0(r_k) = \frac{\rho(r_k)}{\sum_i \rho(r_i)}.$$

Associated with each reward $r_k \in \mathcal{R}_\rho$ and each $x \in \mathcal{X}$, we define the *set of greedy actions at* x with respect to r_k as

$$\mathcal{A}_k(x) = \{a \in \mathcal{A} \mid a \in \operatorname{argmax} Q_k(x,a)\}$$

where Q_k is the Q-function associated with the optimal policy for r_k, as defined in (1). From the sets $\mathcal{A}_k(x), x \in \mathcal{X}$, we define the *greedy policy* with respect to r_k as the mapping $\pi_k : \mathcal{X} \times \mathcal{A} \to [0,1]$ given by

$$\pi_k(x,a) = \frac{\mathbb{I}_{\mathcal{A}_k(x)}(a)}{|\mathcal{A}_k(x)|},$$

where we write $\mathbb{I}_U$ to denote the indicator function for a set U. In other words, for each $x \in \mathcal{X}$, the greedy policy with respect to r_k is defined as a probability distribution that is uniform in $\mathcal{A}_k(x)$ and zero in its complement. We assume, without loss of generality, that for any $r_i, r_j \in \mathcal{R}_\rho$, $\mathcal{A}_i(x) \neq \mathcal{A}_j(x)$ for at least one $x \in \mathcal{X}$.[2]

For any $r_k \in \mathcal{R}_\rho$, consider a perturbed version of π_k where, for each $x \in \mathcal{X}$, action $a \in \mathcal{A}$ is selected with a probability

$$\hat{\pi}_k(x,a) = \begin{cases} \beta_k(x) & \text{if } a \notin \mathcal{A}_k(x) \\ \gamma_k(x) & \text{if } a \in \mathcal{A}_k(x), \end{cases} \tag{2}$$

where, typically, $\beta_k(x) < \gamma_k(x)$.[3] We note that both π_k and the uniform policy can be obtained as limits of $\hat{\pi}_k$, by setting $\beta_k(x) = 0$ or $\beta_k(x) = \gamma_k(x)$, respectively. Following the Bayesian IRL paradigm, the likelihood of observing an action a by the demonstrator at state x, given that the target task is r_k, is now given by

$$\ell_k(x,a) = \mathbb{P}[A_t = a \mid X_t = x, r^* = r_k] = \hat{\pi}_k(x,a). \tag{3}$$

[2] This assumption merely ensures that there are no redundant rewards on $\mathcal{R}_\rho$. If two such rewards r_i, r_j existed in $\mathcal{R}_\rho$, we could safely discard one of the two, say r_j, setting $p_0(r_i) \leftarrow p_0(r_i) + p_0(r_j)$.

[3] Policy $\hat{\pi}_k$ assigns the same probability, $\gamma_k(x)$ to all actions that, for the particular reward r_k, are optimal in state x. Similarly, it assigns the same probability, $\beta_k(x)$, to all corresponding sub-optimal actions. This perturbed version of π_k is convenient both for its simplicity and because it facilitates our analysis. However, other versions of perturbed policies have been considered in the IRL literature—see, for example, the works of Ramachandran and Amir (2007); Neu and Szepesvari (2007); Lopes et al (2009b).

Given a history of t (independent) observations, $\mathcal{F}_t = \{(x_\tau, a_\tau), \tau = 0, \dots, t\}$, the likelihood in (3) can now be used in a standard Bayesian update to compute, for every $r_k \in \mathcal{R}_\rho$, the posterior probability

$$p_t(r_k) = \frac{\mathbb{P}\left[r^* = r_k\right] \mathbb{P}\left[\mathcal{F}_t \mid r^* = r_k\right]}{Z} = \frac{p_0(r_k) \prod_{\tau=0}^{t} \ell_k(x_\tau, a_\tau)}{Z},$$

where Z is a normalization constant.

For the particular case of r^* we write the corresponding perturbed policy as

$$\hat{\pi}^*(x, a) = \begin{cases} \beta^*(x) & \text{if } a \notin \mathcal{A}^*(x) \\ \gamma^*(x) & \text{if } a \in \mathcal{A}^*(x), \end{cases}$$

and denote the *maximum noise level* as the positive constant α defined as

$$\alpha = \sup_{x \in \mathcal{X}} \beta^*(x).$$

3 Multiclass Active Learning for Inverse Reinforcement Learning

In this section we introduce our active learning approach to IRL.

3.1 Preliminaries

To develop an active learning algorithm for this setting, we convert the problem of determining r^* into an equivalent classification problem. This mostly amounts to rewriting of the Bayesian IRL problem from Section 2 using a different notation.

We define the hypothesis space $\mathcal{H}$ as follows. For every $r_k \in \mathcal{R}_\rho$, the kth hypothesis $\mathbf{h}_k : \mathcal{X} \to \{-1, 1\}^{|\mathcal{A}|}$ is defined as the function

$$h_k(x, a) = 2\mathbb{I}_{\mathcal{A}_k(x)}(a) - 1,$$

where we write $h_k(x, a)$ to denote the ath component of $\mathbf{h}_k(x)$. Intuitively, $\mathbf{h}_k(x)$ identifies (with a value of 1) the greedy actions in x with respect to r_k, assigning a value of -1 to all other actions. We take $\mathcal{H}$ as the set of all such functions $\mathbf{h}_k$. Note that, since every reward prescribes at least one optimal action per state, it holds that for every $\mathbf{h} \in \mathcal{H}$ and every $x \in \mathcal{X}$ there is at least one $a \in \mathcal{A}$ such that $h(x, a) = 1$. We write $\mathbf{h}^*$ to denote the target hypothesis, corresponding to r^*.

As before, given a hypothesis $\mathbf{h} \in \mathcal{H}$, we define the *set of greedy actions at* x according to $\mathbf{h}$ as

$$\mathcal{A}_\mathbf{h}(x) = \{a \in \mathcal{A} \mid h(x, a) = 1\}.$$

For an indexed set of samples, $\{(x_\lambda, a_\lambda), \lambda \in \Lambda\}$, we write h_λ to denote $h(x_\lambda, a_\lambda)$, when the index set is clear from the context.

The prior distribution p_0 over $\mathcal{R}_\rho$ induces an equivalent distribution over $\mathcal{H}$, which we abusively also denote as p_0, and is such that $p_0(\mathbf{h}_k) = p_0(r_k)$. We let the history of observations up to time-step t be

$$\mathcal{F}_t = \{(x_\tau, a_\tau), \tau = 0, \dots, t\},$$

and $\beta_\mathbf{h}$ and $\gamma_\mathbf{h}$ be the estimates of β^* and γ^* associated with the hypothesis $\mathbf{h}$. Then, the distribution over $\mathcal{H}$ after observing $\mathcal{F}_t$ can be updated using Bayes rule as

$$\begin{aligned} p_t(\mathbf{h}) &\triangleq \mathbb{P}\left[\mathbf{h}^* = \mathbf{h} \mid \mathcal{F}_t\right] \\ &\propto \mathbb{P}\left[a_t \mid x_t, \mathbf{h}^* = \mathbf{h}, \mathcal{F}_{t-1}\right] \mathbb{P}\left[\mathbf{h}^* = \mathbf{h} \mid \mathcal{F}_{t-1}\right] \\ &= \mathbb{P}\left[a_t \mid x_t, \mathbf{h}^* = \mathbf{h}\right] \mathbb{P}\left[\mathbf{h} = \mathbf{h}^* \mid \mathcal{F}_{t-1}\right] \\ &\approx \gamma_\mathbf{h}(x_t)^{(1+h_t)/2} \beta_\mathbf{h}(x_t)^{(1-h_t)/2} p_{t-1}(\mathbf{h}), \end{aligned} \tag{4}$$

where we assume, for all $x \in \mathcal{X}$,

$$|\mathcal{A}_\mathbf{h}(x)|\, \gamma_\mathbf{h}(x) \leq |\mathcal{A}^*(x)|\, \gamma^*(x), \tag{5}$$

and $p_t(\mathbf{h})$ is normalized so that $\sum_{\mathbf{h} \in \mathcal{H}} p_t(\mathbf{h}) = 1$. Note that, in (4), we accommodate for the possibility of having access (for each hypothesis) to inaccurate estimates $\beta_\mathbf{h}$ and $\gamma_\mathbf{h}$ of β^* and γ^*, respectively.

We consider a partition of the state-space $\mathcal{X}$ into a disjoint family of N sets, $\Xi = \{\mathcal{X}_1, \dots, \mathcal{X}_N\}$ such that all hypotheses $\mathbf{h} \in \mathcal{H}$ are constant in each set $\mathcal{X}_i, i = 1 \dots, N$. In other words, any two states x, y lying in the same $\mathcal{X}_i$ are indistinguishable, since $h(x, a) = h(y, a)$ for all $a \in \mathcal{A}$ and all $\mathbf{h} \in \mathcal{H}$. This means that our hypothesis space $\mathcal{H}$ induces an equivalence relation in $\mathcal{X}$ in which two elements $x, y \in \mathcal{X}$ are equivalent if $\{x, y\} \subset \mathcal{X}_i$. We write $[x]_i$ to denote the (any) representative of the set $\mathcal{X}_i$.[4]

The following definitions extend those of Nowak (2011).

Definition 1 (k-neighborhood)**.** *Two sets $\mathcal{X}_i, \mathcal{X}_j \in \Xi$ are said to be* k-neighbors *if the set*

$$\left\{\mathbf{h} \in \mathcal{H} \mid \mathcal{A}_\mathbf{h}([x]_i) \neq \mathcal{A}_\mathbf{h}([x]_j)\right\}$$

has, at most, k elements, i.e., if there are k or fewer hypotheses in $\mathcal{H}$ that output different optimal actions in $\mathcal{X}_i$ and $\mathcal{X}_j$.

[4] While this partition is, perhaps, of little relevance in problems with a small state-space $\mathcal{X}$, it is central in problems with large (or infinite) state-space, since the state to be queried has to be selected from a set of N alternatives, instead of the (much larger) set of $|\mathcal{X}|$ alternatives.

Definition 2. *The pair* $(\mathcal{X}, \mathcal{H})$ *is* k-neighborly *if, for any two sets* $\mathcal{X}_i, \mathcal{X}_j \in \Xi$*, there is a sequence* $\{\mathcal{X}_{\ell_0}, \ldots, \mathcal{X}_{\ell_n}\} \subset \Xi$ *such that*

- $\mathcal{X}_{\ell_0} = \mathcal{X}_i$ *and* $\mathcal{X}_{\ell_n} = \mathcal{X}_j$*;*
- *For any* m*,* $\mathcal{X}_{\ell_m}$ *and* $\mathcal{X}_{\ell_{m+1}}$ *are* k*-neighbors.*

The notion of k-neighborhood structures the state-space $\mathcal{X}$ in terms of the hypotheses space $\mathcal{H}$, and this structure can be exploited for active learning purposes.

3.2 Active IRL using GBS

In defining our active IRL algorithm, we first consider a simplified setting in which the following assumption holds. We postpone to Section 3.3 the discussion of the more general case.

Assumption 1. *For every* $h \in \mathcal{H}$ *and every* $x \in \mathcal{X}$*,* $|\mathcal{A}_{\mathbf{h}}(x)| = 1$.

In other words, we focus on the case where all hypothesis considered prescribe a unique optimal action per state. A single optimal action per state implies that the noise model can be simplified. In particular, the noise model can now be constant across hypothesis, since all $\mathbf{h} \in \mathcal{H}$ prescribes the same number of optimal actions in each state (namely, one). We denote by $\hat{\gamma}(x)$ and $\hat{\beta}(x)$ the estimates of γ^* and β^*, respectively, and consider a Bayesian update of the form:

$$p_t(\mathbf{h}) \propto \frac{1}{Z}\hat{\gamma}(x_t)^{(1+h_t)/2}\hat{\beta}(x_t)^{(1-h_t)/2}p_{t-1}(\mathbf{h}), \qquad (6)$$

with $1 - \hat{\gamma}(x) = (|\mathcal{A}| - 1)\hat{\beta}(x)$ and Z an adequate normalization constant. For this simpler case, (5) becomes

$$\hat{\beta}(x) \geq \beta^*(x) \qquad \text{and} \qquad \hat{\gamma}(x) \leq \gamma^*(x), \qquad (7)$$

where, as before, we overestimate the noise rate $\beta^*(x)$. For a given probability distribution p, define the *weighted prediction in* x as

$$W(p, x) = \max_{a \in \mathcal{A}} \sum_{\mathbf{h} \in \mathcal{H}} p(\mathbf{h}) h(x, a),$$

and the *predicted action at* x as

$$A^*(p, x) = \operatorname*{argmax}_{a \in \mathcal{A}} \sum_{\mathbf{h} \in \mathcal{H}} p(\mathbf{h}) h(x, a).$$

We are now in position to introduce a first version of our active learning algorithm for inverse reinforcement learning, that we dub *Generalized Binary Search for IRL* (GBS-IRL). GBS-IRL is summarized in Algorithm 1. This first version of the algorithm relies critically on Assumption 1. In Section 3.3, we discuss how Algorithm 1 can be modified to accommodate situations in which Assumption 1 does not hold.

Algorithm 1 GBS-IRL (version 1)

Require: MDP parameters $\mathcal{M} \backslash r$
Require: Reward space $\mathcal{R}_\rho$
Require: Prior distribution p_0 over $\mathcal{R}$
Compute $\mathcal{H}$ from $\mathcal{R}_\rho$
Determine partition $\Xi = \mathcal{X}_1, \ldots \mathcal{X}_N$ of $\mathcal{X}$
Set $\mathcal{F}_0 = \emptyset$
for all $t = 0, \ldots$ **do**
  Set $c_t = \min_{i=1,\ldots,N} W(p_t, [x]_i)$
  if there are 1-neighbor sets $\mathcal{X}_i, \mathcal{X}_j$ such that

$$W(p_t, [x]_i) > c_t, \qquad W(p_t, [x]_j) > c_t$$
$$A^*(p_t, [x]_i) \neq A^*(p_t, [x]_j),$$

  then
    Sample x_{t+1} from $\mathcal{X}_i$ or $\mathcal{X}_j$ with probability 1/2
  else
    Sample x_{t+1} from the set $\mathcal{X}_i$ that minimizes $W(p_t, [x]_i)$.
  end if
  Obtain noisy response a_{t+1}
  Set $\mathcal{F}_{t+1} \leftarrow \mathcal{F}_t \cup \{(x_{t+1}, a_{t+1})\}$
  Update p_{t+1} from p_t using (6)
end for
return $\hat{\mathbf{h}}_t = \operatorname{argmax}_{\mathbf{h} \in \mathcal{H}} p_t(\mathbf{h})$.

Our analysis of GBS-IRL relies on the following fundamental lemma that generalizes Lemma 3 of Nowak (2011) to multi-class settings.

Lemma 1. *Let* $\mathcal{H}$ *denote a hypothesis space defined over a set* $\mathcal{X}$*, where* $(\mathcal{X}, \mathcal{H})$ *is assumed* k*-neighborly. Define the* coherence parameter *for* $(\mathcal{X}, \mathcal{H})$ *as*

$$c^*(\mathcal{X}, \mathcal{H}) \triangleq \max_{a \in \mathcal{A}} \min_{\mu} \max_{\mathbf{h} \in \mathcal{H}} \sum_{i=1}^{N} h([x]_i, a)\mu(\mathcal{X}_i),$$

where μ *is a probability measure over* $\mathcal{X}$*. Then, for any probability distribution* p *over* $\mathcal{H}$*, one of the two statements below holds:*

1. *There is a set* $\mathcal{X}_i \in \Xi$ *such that*

$$W(p, [x]_i) \leq c^*.$$

2. *There are two* k*-neighbor sets* $\mathcal{X}_i$ *and* $\mathcal{X}_j$ *such that*

$$W(p, [x]_i) > c^* \qquad W(p, [x]_j) > c^*$$
$$A^*(p, [x]_i) \neq A^*(p, [x]_j).$$

Proof. See Appendix A.1. □

This lemma states that, given any distribution over the set of hypothesis, either there is a state $[x]_i$ for

which there is great uncertainty concerning the optimal action or, alternatively, there are two k-neighboring states $[x]_i$ and $[x]_j$ in which all except a few hypothesis predict the same action, yet the predicted optimal action is strikingly different in both states. In either case, it is possible to select a query that is highly informative.

The coherence parameter c^* is the multi-class equivalent of the coherence parameter introduced by Nowak (2011), and quantifies the informativeness of queries. That c^* always exists can be established by noting that the partition of $\mathcal{X}$ is finite (since $\mathcal{H}$ is finite) and, therefore, the minimization can be conducted exactly. On the other hand, if $\mathcal{H}$ does not include trivial hypotheses that are constant all over $\mathcal{X}$, it holds that $c^* < 1$.

We are now in position to establish the convergence properties of Algorithm 1. Let $\mathbb{P}[\cdot]$ and $\mathbb{E}[\cdot]$ denote the probability measure and corresponding expectation governing the underlying probability over noise and possible algorithm randomizations in query selection.

Theorem 1 (Consistency of GBS-IRL)**.** *Let $\mathcal{F}_t = \{(x_\tau, a_\tau), \tau = 1, \dots, t\}$ denote a possible history of observations obtained with GBS-IRL. If, in the update* (6)*, $\hat{\beta}(x)$ and $\hat{\gamma}(x)$ verify* (7)*, then*

$$\lim_{t\to\infty} \mathbb{P}\big[\hat{\mathbf{h}}_t \neq h^*\big] = 0.$$

Proof. See Appendix A.2. □

Theorem 1 establishes the consistency of active learning for multi-class classification. The proof relies on a fundamental lemma that, roughly speaking, ensures that the sequence $p_t(\mathbf{h}^*)$ is increasing in expectation. This fundamental lemma (Lemma 2 in Appendix A.2) generalizes a related result of Nowak (2011) that, due to the consideration of multiple classes in GBS-IRL, does not apply. Our generalization requires, in particular, stronger assumptions on the noise, $\hat{\beta}(x)$, and implies a different rate of convergence, as will soon become apparent. It is also worth mentioning that the statement in Theorem 1 could alternatively be proved using an adaptive sub-modularity argument (again relying on Lemma 2 in Appendix A.2), using the results of Golovin and Krause (2011).

Theorem 1 ensures that, as the number of samples increases, the probability mass concentrates on the correct hypothesis $\mathbf{h}^*$. However, it does not provide any information concerning the rate at which $\mathbb{P}\big[\hat{\mathbf{h}}_t \neq h^*\big] \to 0$. The convergence rate for our active sampling approach is established in the following result.

Theorem 2 (Convergence Rate of GBS-IRL)**.** *Let $\mathcal{H}$ denote our hypothesis space, defined over $\mathcal{X}$, and assume that $(\mathcal{X}, \mathcal{H})$ is 1-neighborly. If, in the update* (6)*, $\hat{\beta}(x) > \alpha$ for all $x \in \mathcal{X}$, then*

$$\mathbb{P}\big[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\big] \leq |\mathcal{H}|\,(1-\lambda)^t, \quad t = 0, \dots \tag{8}$$

where $\lambda = \varepsilon \cdot \min\left\{\frac{1-c^*}{2}, \frac{1}{4}\right\} < 1$ and

$$\varepsilon = \min_x \gamma^*(x)\frac{\hat{\gamma}(x) - \hat{\beta}(x)}{\hat{\gamma}(x)} + \beta^*(x)\frac{\hat{\beta}(x) - \hat{\gamma}(x)}{\hat{\beta}(x)}. \tag{9}$$

Proof. See Appendix A.4. □

Theorem 2 extends Theorem 4 of Nowak (2011) to the multi-class case. However, due to the existence of multiple actions (classes), the constants obtained in the above bounds differ from those obtained in the aforementioned work (Nowak, 2011). Interestingly, for c^* close to zero, the convergence rate obtained is near-optimal, exhibiting a logarithmic dependence on the dimension of the hypothesis space. In fact, we have the following straightforward corollary of Theorem 2.

Corollary 1 (Sample Complexity of GBS-IRL)**.** *Under the conditions of Theorem 2, for any given $\delta > 0$, $\mathbb{P}\big[\hat{\mathbf{h}}_t = \mathbf{h}^*\big] > 1 - \delta$ as long as*

$$t \geq \frac{1}{\lambda} \log \frac{|\mathcal{H}|}{\delta}.$$

To conclude this section, we note that our reduction of IRL to a standard (multi-class) classification problem implies that Algorithm 1 is not specialized in any particular way to IRL problems—in particular, it can be used in general classification problems. Additionally, the guarantees in Theorems 1 and 2 are also generally applicable in any multi-class classification problems verifying the corresponding assumptions.

3.3 Discussion and Extensions

We now discuss the general applicability of our results from Section 3.2. In particular, we discuss two assumptions considered in Theorem 2, namely the 1-neighborly condition on $(\mathcal{X}, \mathcal{H})$ and Assumption 1. We also discuss how additional forms of expert feedback may be integrated in a seamless manner in our GBS-IRL approach, so that the learner is able to recover efficiently the target task.

1-Neighborly Assumption:

This assumption is formulated in Theorem 2. The 1-neighborly assumption states that $(\mathcal{X}, \mathcal{H})$ is 1-neighborly, meaning that it is possible to "structure" the state-space $\mathcal{X}$ in a manner that is coherent with the hypothesis space $\mathcal{H}$. To assess the validity of this assumption in general, we start by recalling that two sets $\mathcal{X}_i, \mathcal{X}_j \in \Xi$ are 1-neighbors if there is a single hypothesis $\mathbf{h}_0 \in \mathcal{H}$ that prescribes different optimal actions in $\mathcal{X}_i$ and $\mathcal{X}_j$. Then, $(\mathcal{X}, \mathcal{H})$ is 1-neighborly if every two sets $\mathcal{X}_i, \mathcal{X}_j$ can be "connected" by a sequence of 1-neighbor sets.

In general, given a multi-class classification problem with hypothesis space $\mathcal{H}$, the 1-neighborly assumption can be investigated by verifying the connectivity of the 1-neighborhood graph induced by $\mathcal{H}$ on $\mathcal{X}$. We refer to the work of Nowak (2011) for a detailed discussion of this case, as similar arguments carry to our multi-class extension.

In the particular case of inverse reinforcement learning, it is important to assess whether the 1-neighborly assumption is reasonable. Given a finite state-space, $\mathcal{X}$, and a finite action-space, $\mathcal{A}$, it is possible to build a total of $|\mathcal{A}|^{|\mathcal{X}|}$ different hypothesis.[5] As shown in the work of Melo et al (2010), for any such hypothesis it is always possible to build a non-degenerate reward function that yields such hypothesis as the optimal policy. Therefore, a sufficiently rich reward space ensures that the corresponding hypothesis space $\mathcal{H}$ includes all $|\mathcal{A}|^{|\mathcal{X}|}$ possible policies already alluded to. This trivially implies that $(\mathcal{X}, \mathcal{H})$ is *not* 1-neighborly.

Unfortunately, as also shown in the aforementioned work (Melo et al, 2010), the consideration of $\mathcal{H}$ as the set of all possible policies also implies that all states must be sufficiently sampled, since no generalization across states is possible. This observation supports the option in most IRL research to focus on problems in which rewards/policies are selected from some restricted set (for example, Abbeel and Ng, 2004; Ramachandran and Amir, 2007; Neu and Szepesvari, 2007; Syed and Schapire, 2008). For the particular case of active learning approaches, the consideration of a full set of rewards/policies also implies that there is little hope that any active sampling will provide any but a negligible improvement in sample complexity. A related observation can be found in the work of Dasgupta (2005) in the context of active learning for binary classification.

[5] This number is even larger if multiple optimal actions are allowed.

Algorithm 2 GBS-IRL (version 2)

Require: MDP parameters $\mathcal{M}\backslash r$
Require: Reward space $\mathcal{R}_\rho$
Require: Prior distribution p_0 over $\mathcal{R}$
Compute $\mathcal{H}$ from $\mathcal{R}_\rho$
Determine partition $\Xi = \mathcal{X}_1, \dots \mathcal{X}_N$ of $\mathcal{X}$
Set $\mathcal{F}_0 = \emptyset$
for all $t = 0, \dots$ **do**
  Sample x_{t+1} from the set $\mathcal{X}_i$ that minimizes $W(p_t, [x]_i)$.
  Obtain noisy response a_{t+1}
  Set $\mathcal{F}_{t+1} \leftarrow \mathcal{F}_t \cup \{(x_{t+1}, a_{t+1})\}$
  Update p_{t+1} from p_t using (6)
end for
return $\hat{\mathbf{h}}_t = \operatorname{argmax}_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})$.

In situations where the 1-neighborly assumption may not be verified, Lemma 1 cannot be used to ensure the selection of highly informative queries once $W(p, [x]_i) > c^*$ for all $\mathcal{X}_i$. However, it should still be possible to use the main approach in GBS-IRL, as detailed in Algorithm 2. For this situation, we can specialize our sample complexity results in the following immediate corollary.

Corollary 2 (Convergence Rate of GBS-IRL, version 2)**.** *Let $\mathcal{H}$ denote our hypothesis space, defined over $\mathcal{X}$, and let $\hat{\beta}(x) > \alpha$ in the update* (6)*. Then, for all t such that $W(p_t, [x]_i) \leq c^*$ for some $\mathcal{X}_i$,*

$$\mathbb{P}\big[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\big] \leq |\mathcal{H}|\,(1-\lambda)^t, \quad t = 0, \dots$$

where $\lambda = \varepsilon\frac{1-c^*}{2}$ and ε is defined in (9)*.*

Multiple Optimal Actions:

In our presentation so far, we assumed that $\mathcal{R}_\rho$ is such that, for any $r \in \mathcal{R}_\rho$ and any $x \in \mathcal{X}$, $|\mathcal{A}_r(x)| = 1$ (Assumption 1). Informally, this corresponds to assuming that, for every reward function considered, there is a single optimal action, $\pi^*(x)$, at each $x \in \mathcal{X}$. This assumption has been considered, either explicitly or implicitly, in several previous works on learning by demonstration (see, for example, the work of Chernova and Veloso, 2009). Closer to our own work on active IRL, several works recast IRL as a classification problem, focusing on deterministic policies $\pi_k : \mathcal{X} \to \mathcal{A}$ (Ng and Russel, 2000; Cohn et al, 2011; Judah et al, 2011; Ross and Bagnell, 2010; Ross et al, 2011) and therefore, although not explicitly, also consider a single optimal action in each state.

However, MDPs with multiple optimal actions per state are not uncommon (the scenarios considered in Section 4, for example, have multiple optimal actions

per state). In this situation, the properties of the resulting algorithm do not follow from our previous analysis, since the existence of multiple optimal actions necessarily requires a more general noise model. The immediate extension of our noise model to a scenario where multiple optimal actions are allowed poses several difficulties, as optimal actions across policies may be sampled with different probabilities.

In order to overcome such difficulty, we consider a more conservative Bayesian update, that enables a seamless generalization of our results to scenarios that admit multiple optimal actions in each state. Our update now arises from considering that the likelihood of observing an action from a set $\mathcal{A}_{\mathbf{h}}(x)$ at state x is given by $\gamma_{\mathbf{h}}(x)$. Equivalently, the likelihood of observing an action from $\mathcal{A} - \mathcal{A}_{\mathbf{h}}(x)$ is given by $\beta_{\mathbf{h}}(x) = 1 - \gamma_{\mathbf{h}}(x)$. As before, γ^* and β^* correspond to the values of $\gamma_{\mathbf{h}}$ and $\beta_{\mathbf{h}}$ for the target hypothesis, and we again let

$$\alpha = \sup_{x \in \mathcal{X}} \beta^*(x).$$

Such *aggregated* noise model again enables the consideration of an approximate noise model that is constant across hypothesis, and is defined in terms of estimates $\hat{\gamma}(x)$ and $\hat{\beta}(x)$ of $\gamma^*(x)$ and $\beta^*(x)$. Given the noise model just described, we get the Bayesian update

$$\begin{aligned} p_t(\mathbf{h}) &\triangleq \mathbb{P}\left[\mathbf{h}^* = \mathbf{h} \mid \mathcal{F}_t\right] \\ &\propto \mathbb{P}\left[a_t \in \mathcal{A}_{\mathbf{h}} \mid x_t, \mathcal{F}_{t-1}\right] \mathbb{P}\left[\mathbf{h}^* = \mathbf{h} \mid \mathcal{F}_{t-1}\right] \\ &= \mathbb{P}\left[a_t \in \mathcal{A}_{\mathbf{h}} \mid x_t\right] \mathbb{P}\left[\mathbf{h} = \mathbf{h}^* \mid \mathcal{F}_{t-1}\right] \\ &\approx \hat{\gamma}(x)^{(1+h_t)/2} \hat{\beta}(x)^{(1-h_t)/2} p_{t-1}(\mathbf{h}), \end{aligned} \tag{10}$$

with $\hat{\gamma}(x)$ and $\hat{\beta}(x)$ verifying (7). This revised formulation implies that the updates to p_t are more conservative, in the sense that they are slower to "eliminate" hypothesis from $\mathcal{H}$. However, all results for Algorithm 1 remain valid with the new values for $\hat{\gamma}$ and $\hat{\beta}$.

Unfortunately, by allowing multiple optimal actions per state, it is also much easier to find (non-degenerate) situations where $c^* = 1$, in which case our bounds are void. However, if we focus on identifying, in each state, at least *one optimal action*, we are able to retrieve some guarantees on the sample complexity of our active learning approach. We thus consider yet another version of GBS-IRL, described in Algorithm 3, that uses uses a threshold $\hat{c} < 1$ such that, if $W(p_t, [x]_i) > \hat{c}$, we consider that (at least) one optimal action at $[x]_i$ has been identified. Once this is done, it outputs the most likely hypothesis. Once at least one optimal action has been identified in all states, the algorithm stops.

Algorithm 3 GBS-IRL (version 3)

Require: MDP parameters $\mathcal{M} \backslash r$
Require: Reward space $\mathcal{R}_\rho$
Require: Prior distribution p_0 over $\mathcal{R}$
Compute $\mathcal{H}$ from $\mathcal{R}_\rho$
Determine partition $\Xi = \mathcal{X}_1, \dots \mathcal{X}_N$ of $\mathcal{X}$
Set $\mathcal{F}_0 = \emptyset$
for all $t = 0, \dots$ **do**
  Set $c_t = \min_{i=1,\dots,N} W(p_t, [x]_i)$
  if $c_t < \hat{c}$ **then**
    Sample x_{t+1} from the set $\mathcal{X}_i$ that minimizes $W(p_t, [x]_i)$.
  else
    Return $\hat{\mathbf{h}}_t = \operatorname{argmax}_{\mathbf{h} \in \mathcal{H}} p_t(\mathbf{h})$.
  end if
  Obtain noisy response a_{t+1}
  Set $\mathcal{F}_{t+1} \leftarrow \mathcal{F}_t \cup \{(x_{t+1}, a_{t+1})\}$
  Update p_{t+1} from p_t using (10)
end for

To analyze the performance of this version of GBS-IRL, let the set of *predicted optimal actions at* x be defined as

$$\mathcal{A}_{\hat{c}}(p, x) = \left\{ a \in \mathcal{A} \mid \sum_{\mathbf{h}} p(\mathbf{h}) h(x, a) > \hat{c} \right\}.$$

We have the following results.

Theorem 3 (Consistency of GBS-IRL, version 3)**.** *Consider any history of observations* $\mathcal{F}_t = \{(x_\tau, a_\tau), \tau = 1, \dots, t\}$ *from GBS-IRL. If, in the update* (6)*,* $\hat{\beta}$ *and* $\hat{\gamma}$ *verify* (7) *for all* $\mathbf{h} \in \mathcal{H}$*, then for any* $a \in \mathcal{A}_{\hat{c}}(p, [x]_i)$*,*

$$\lim_{t \to \infty} \mathbb{P}\big[h^*([x]_i, a) \neq 1\big] = 0.$$

Proof. See Appendix A.5. □

Note that the above result is no longer formulated in terms of the identification of the correct hypothesis, but in terms of the identification of the set of optimal actions. We also have the following result on the sample complexity of version 3 of GBS-IRL.

Corollary 3 (Convergence Rate of GBS-IRL, version 3)**.** *Let* $\mathcal{H}$ *denote our hypothesis space, defined over* $\mathcal{X}$*, and let* $\hat{\beta}(x) > \alpha$ *in the update* (10)*. Then, for all* t *such that* $W(p_t, [x]_i) \leq c^*$ *for some* $\mathcal{X}_i$*, and all* $a \in \mathcal{A}_{\hat{c}}(p, [x]_i)$*,*

$$\mathbb{P}\big[h^*([x]_i, a) \neq 1\big] \leq |\mathcal{H}| \, (1 - \lambda)^t, \quad t = 0, \dots$$

where $\lambda = \varepsilon \frac{1 - c^*}{2}$ *and* ε *is defined in* (9) *with the new values for* $\hat{\gamma}$ *and* $\hat{\beta}$*.*

Different Query Types:

Finally, it is worth noting that, in the presentation so far admits for queries such as *"What is the optimal action in state x?"* However, it is possible to devise different types of queries (such as *"Is action a optimal in state x?"*) that enable us to recover the stronger results in Theorem 2. In fact, a query such as the one exemplified reduces the IRL problem to a *binary classification problem* over $\mathcal{X} \times \mathcal{A}$, for which existing active learning methods such as the one of Nowak (2011) can readily be applied.

Integrating Reward Feedback:

So far, we discussed one possible approach to IRL, where the agent is provided with a demonstration $\mathcal{F}_t = \{(x_\tau, a_\tau), \tau = 1, \ldots, t\}$ consisting of pairs (x_τ, a_τ) of states and corresponding actions. From this demonstration the agent must identify the underlying target task, represented as a reward function, r^*. We now depart from the Bayesian formalism introduced above and describe how reward information can also be integrated.

With the addition of reward information, our demonstrations may now include state-reward pairs (x_τ, u_τ), indicating that the reward in state x_τ takes the value u_τ. This can be seen as a similar approach as those of Thomaz and Breazeal (2008); Knox and Stone (2010) for reinforcement learning. The main difference is that, in the aforementioned works, actions are experienced by the learner who then receives rewards both from the environment and the teacher. Another related approach is introduced by Regan and Boutilier (2011), in the context of reward design for MDPs.

As with action information, the demonstrator would ideally provide exact values for r^*. However, we generally allow the demonstration to include some level of noise, where

$$\mathbb{P}\left[u_\tau = u \mid x_\tau, r^*\right] \propto e^{(u - r_{\text{target}}(x))^2/\sigma}, \qquad (11)$$

where σ is a non-negative constant. As with policy information, reward information can be used to update $p_t(r_k)$ as

$$\begin{aligned} p_t(r_k) &\triangleq \mathbb{P}\left[r^* = r_k \mid \mathcal{F}_t\right] \\ &\propto \mathbb{P}\left[u_t \mid x_t, r^* = r_k, \mathcal{F}_{t-1}\right] \mathbb{P}\left[r^* = r_k \mid \mathcal{F}_{t-1}\right] \\ &= \mathbb{P}\left[u_t \mid x_t, r^* = r_k\right] \mathbb{P}\left[r^* = r_k \mid \mathcal{F}_{t-1}\right] \\ &\approx e^{(u_t - r_k(x))^2/\hat{\sigma}} p_{t-1}(r_k) \end{aligned}$$

where, as before, we allow for an inaccurate estimate $\hat{\sigma}$ of σ such that $\hat{\sigma} \geq \sigma$. Given the correspondence between the rewards in $\mathcal{R}_\rho$ and the hypothesis in $\mathcal{H}$, the above Bayesian update can be used to seamlessly integrate reward information in our Bayesian IRL setting.

To adapt our active learning approach to accommodate for reward feedback, let

$$x_{t+1} = \operatorname*{argmin}_{\mathcal{X}_i, i=1,\ldots,N} W(p_t, [x]_i).$$

i.e., x_{t+1} is the state that would be queried by Algorithm 1 at time-step $t+1$. If the user instead wishes to provide reward information, we would like to replace the query x_{t+1} by some alternative query x'_{t+1} that disambiguates as much as possible the actions in state x_{t+1}—much like a direct query to x_{t+1} would.

To this purpose, we partition the space of rewards, $\mathcal{R}_\rho$, into $|\mathcal{A}|$ or less disjoint sets $\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{A}|}$, where each set $\mathcal{R}_a$ contains precisely those rewards $r \in \mathcal{R}_\rho$ for which $\pi_r(x_{t+1}) = a$. We then select the state $x'_{t+1} \in \mathcal{X}$, the reward at which best discriminates between the sets $\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{A}|}$. The algorithm will then query the demonstrator for the reward at this new state.

In many situations, the rewards in $\mathcal{R}_\rho$ allow only poor discrimination between the sets $\mathcal{R}_1, \ldots, \mathcal{R}_{|\mathcal{A}|}$. This is particularly evident if the reward is sparse, since after a couple informative reward samples, all other states contain similar reward information. In Section 4 we illustrate this inconvenience, comparing the performance of our active method in the presence of both sparse and dense reward functions.

4 Experimental Results

This section illustrates the application of GBS-IRL in several problems of different complexity. It also features a comparison with other existing methods from the active IRL literature.

4.1 GBS-IRL

In order to illustrate the applicability of our proposed approach, we conducted a series of experiments where GBS-IRL is used to determine the (unknown) reward function for some underlying MDP, given a perturbed demonstration of the corresponding policy.

In each experiment, we illustrate and discuss the performance of GBS-IRL. The results presented correspond to averages over 200 independent Monte-Carlo trials, where each trial consists of a run of 100 learning steps, in each of which the algorithm is required to select one state to query and is provided the corresponding action. GBS-IRL is initialized with a set $\mathcal{R}_\rho$ of 500 independently generated random rewards.

This set always includes the correct reward, r^* and the remaining rewards are built to have similar range and sparsity as that of r^*.

The prior probabilities, $p_0(r)$, are proportional to the level of sparsity of each reward r. This implies that some of the random rewards in $\mathcal{R}_\rho$ may have larger prior probability than r^*. For simplicity, we considered an exact noise model, *i.e.*, $\hat{\beta} = \beta^*$ and $\hat{\gamma} = \gamma^*$, where $\beta^*(x) \equiv 0.1$ and $\gamma^*(x) \equiv 0.9$, for all $x \in \mathcal{X}$.

For comparison purposes, we also evaluated the performance of other active IRL approaches from the literature, to know:

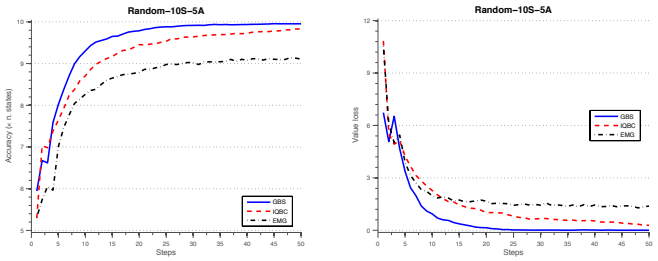
- The *imitation query-by-committee* algorithm (IQBC) of Judah et al (2011), that uses an entropy-based criterion to select the states to query.
- The *expected myopic gain* algorithm (EMG) of Cohn et al (2011), that uses a criterion based on the expected gain of information to select the states to query.

As pointed out in Section 1.1, IQBC is, in its core, very similar to GBS-IRL, the main differences being in terms of the selection criterion and of the fact that the IQBC is able to accommodate the notion of "bad states". Since this notion is not used in our examples, we expect the performance of both methods to be essentially similar.

As for EMG, this algorithm queries the expert for the correct action in those states where the expected gain of information is potentially larger (Cohn et al, 2011). This requires evaluating, for each state $x \in \mathcal{X}$ and each possible outcome, the associated gain of information. Such method is, therefore, fundamentally different from GBS-IRL and we expect this method to yield crisper differences from our own approach. Additionally, the above estimation is computationally heavy, as (in the worst case) requires the evaluation of an MDP policy for each state-action pair.

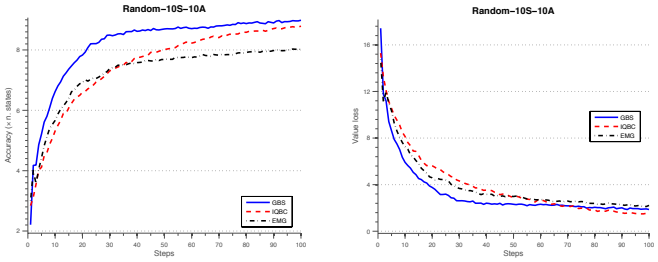
Small-sized random MDPs

In the first set of experiments, we evaluate the performance of GBS-IRL in several small-sized MDPs with no particular structure (both in terms of transitions and in terms of rewards). Specifically, we considered MDPs where $|\mathcal{X}| = 10$ and either $|\mathcal{A}| = 5$ or $|\mathcal{A}| = 10$. For each MDP size, we consider 10 random and independently generated MDPs, in each of which we conducted 200 independent learning trials. This first set of experiments serves two purposes. On one hand, it illustrates the applicability of GBS-IRL in arbitrary settings, by evaluating the performance of our method in random MDPs with no particular structure. On the other hand, these initial experiments also enable a quick comparative analysis of GBS-IRL against other relevant methods from the active IRL literature.



(a) (b)

Figure 1: Performance of all methods in random MDPs with $|\mathcal{X}| = 10$ and $|\mathcal{A}| = 5$.



(a) (b)

Figure 2: Performance of all methods in random MDPs with $|\mathcal{X}| = 10$ and $|\mathcal{A}| = 10$.

Figures 1(a) and 2(a) depict the learning curve for all three methods in terms of policy accuracy. The performance of all three methods is essentially similar in the early stages of the learning process. However, GBS-IRL slightly outperforms the other two methods, although the differences from IQBC are, as expected, smaller than those from EMG.

While policy accuracy gives a clear view of the learning performance of the algorithms, it conveys a less clear idea on the ability of the learned policies to complete the task intended by the demonstrator. To evaluate the performance of the three learning algorithms in terms of the target task, we also measured the loss of the learned policies with respect to the optimal policy. Results are depicted in Figs. 1(b) and 2(b). These results also confirm that the performance of GBS-IRL is essentially similar. In particular, the differences observed in terms of policy accuracy have little impact in terms of the ability to perform the target task compe-

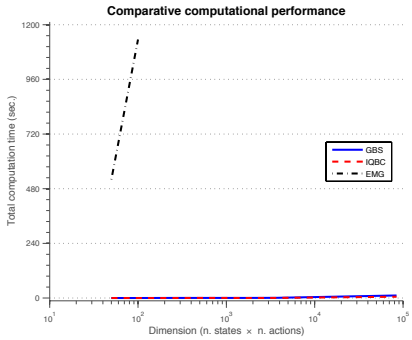


Figure 3: Average (total) computational time for problems of different dimensions.

tently.

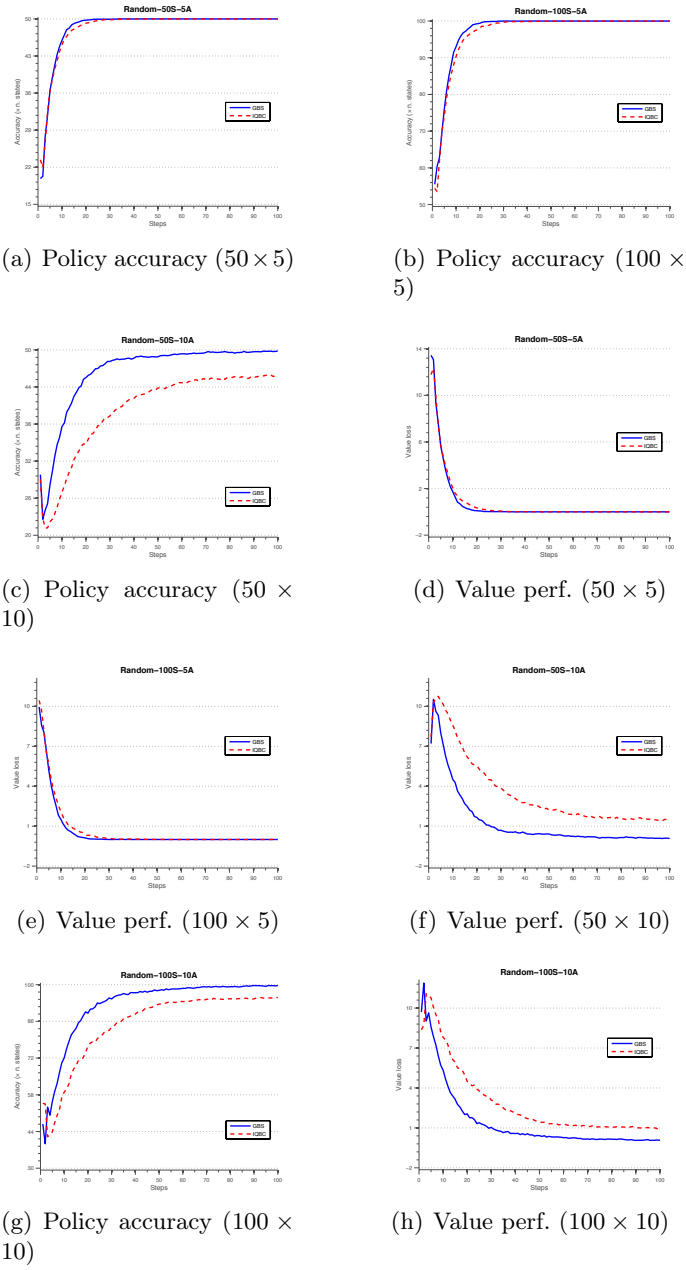
To conclude this section, we also compare the computation time for all methods in these smaller problems. The results are depicted in Fig. 3. We emphasize that the results portrayed herein are only indicative, as all algorithms were implemented in a relatively straightforward manner, with no particular concerns for optimization. Still, the comparison does confirm that the computational complexity associated with EMG is many times superior to that involved in the remaining methods. This, discussed earlier, is due to the heavy computations involved in the estimation of the expected myopic gain, which grows directly with the size of $|\mathcal{X}| \times |\mathcal{A}|$. This observation is also in line with the discussion already found in the original work of Cohn et al (2011).

Medium-sized random MDPs

In the second set of experiments, we investigate how the performance of GBS-IRL is affected by the dimension of the domain considered. To this purpose, we evaluate the performance of GBS-IRL in arbitrary medium-sized MDPs with no particular structure (both in terms of transitions and in terms of rewards). Specifically, we now consider MDPs where either $|\mathcal{X}| = 50$ or $|\mathcal{X}| = 100$, and again take either $|\mathcal{A}| = 5$ or $|\mathcal{A}| = 10$. For each MDP size, we consider 10 random and independently generated MDPs, in each of which we conducted 200 independent learning trials.

Given the results in the first set of experiments and the computation time already associated with EMG, in the remaining experiments we opted by comparing GBS-IRL with IQBC only. The learning curves in terms both of policy accuracy and task execution are depicted in Fig. 4.

In this set of experiments we can observe that the performance of IQBC appears to deteriorate more



(a) Policy accuracy (50×5)

(b) Policy accuracy (100×5)

(c) Policy accuracy (50×10)

(d) Value perf. (50×5)

(e) Value perf. (100×5)

(f) Value perf. (50×10)

(g) Policy accuracy (100×10)

(h) Value perf. (100×10)

Figure 4: Classification and value performance of GBS-IRL and IQBC in medium-sized random MDPs. Solid lines correspond to GBS-IRL, and dotted lines correspond to IQBC. (a)-(g) Classification performance. (d)-(h) Value performance. The indicated values correspond to the dimensions $|\mathcal{X}| \times |\mathcal{A}|$ of the MDPs.

severely with the number of actions than that of GBS-IRL. Although not significantly, this tendency could already be observed in the smaller environments (see, for example, Fig. 2(b)). This dependence on the number of actions is not completely unexpected. In fact,

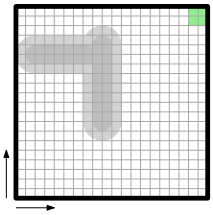


Figure 5: The puddle-world domain (Boyan and Moore, 1995).

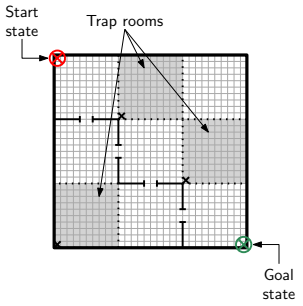


Figure 6: The trap-world domain (Judah et al, 2011).

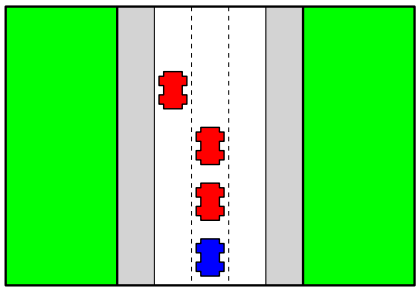


Figure 7: The driver-world domain (Abbeel and Ng, 2004).

IQBC queries states x that maximize

$$VE(x) = -\sum_{a\in\mathcal{A}} \frac{n_{\mathcal{H}}(x,a)}{|\mathcal{H}|} \log \frac{n_{\mathcal{H}}(x,a)}{|\mathcal{H}|},$$

where $n_{\mathcal{H}}(x,a)$ is the number of hypothesis $\mathbf{h} \in \mathcal{H}$ such that $a \in \mathcal{A}_{\mathbf{h}}(x)$. Since the disagreement is taken over the set of all possible actions, there is some dependence of the performance of IQBC on the number of actions.

GBS-IRL, on the other hand, is more focused toward identifying *one* optimal action per state. This renders our approach less sensitive to the number of actions, as can be seen in Corollaries 1 through 3 and illustrated in Fig. 4.

Large-sized structured domains

So far, we have analyzed the performance of GBS-IRL in random MDPs with no particular structure, both in terms of transition probabilities and reward function. In the third set of experiments, we look further into the scalability of GBS-IRL by considering large-sized domains. We consider more structured problems selected from the IRL literature. In particular, we evaluate the performance of GBS-IRL in the *trap-world*, *puddle-world* and *driver* domains.

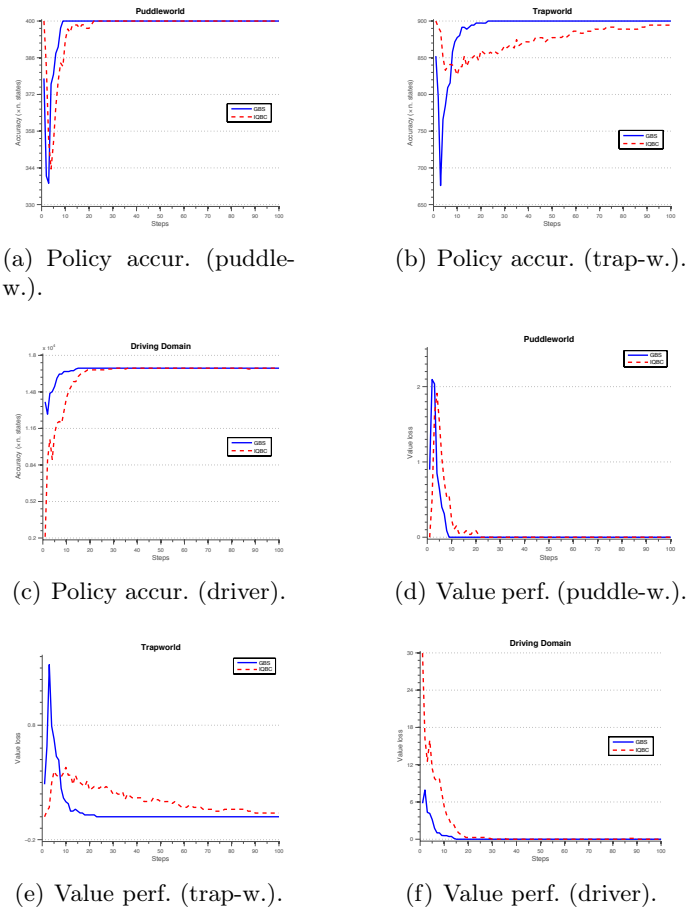
The *puddle-world* domain was introduced in the work of Boyan and Moore (1995), and is depicted in Fig. 5. It consists of a 20×20 grid-world in which two "puddles" exist (corresponding to the darker cells). When in the puddle, the agent receives a penalty that is proportional to the squared distance to the nearest edge of the puddle, and ranges between 0 and -1. The agent must reach the goal state in the top-right corner of the environment, upon which it receives a reward of $+1$. We refer to the original description of Boyan and Moore (1995) for further details.

This domain can be described by an MDP with $|\mathcal{X}| = 400$ and $|\mathcal{A}| = 4$, where the four actions correspond to motion commands in the four possible directions. Transitions are stochastic, and can be described as follows. After selecting the action corresponding to moving in direction d, the agent will roll back one cell (*i.e.,* move in the direction $-d$) with a probability 0.06. With a probability 0.24 the action will fail and the agent will remain in the same position. The agent will move to the adjacent position in direction d with probability 0.4. With a probability 0.24 it will move two cells in direction d, and with probability 0.06 it will move three cells in direction d. We used a discount $\gamma = 0.95$ for the MDP (not to be confused with the noise parameters, $\hat{\gamma}(x)$).

The *trap-world* domain was introduced in the work of Judah et al (2011), and is depicted in Fig. 6. It consists of a 30×30 grid-world separated into 9 rooms. Darker rooms correspond to *trap rooms*, from which the agent can only leave by reaching the corresponding bottom-left cell (marked with a "$\times$"). Dark lines correspond to walls that the agent cannot traverse. Dotted lines are used to delimit the trap-rooms from the safe rooms but are otherwise meaningless. The agent must reach the goal state in the bottom-right corner of the environment. We refer to the work of Judah et al (2011) for a more detailed description.

This domain can be described by an MDP with $|\mathcal{X}| = 900$ and $|\mathcal{A}| = 4$, where the four actions correspond to motion commands in the four possible directions. Transitions are deterministic. The target reward function r^* is everywhere 0 except on the goal, where $r^*(x_{\text{goal}}) = 1$. We again used a discount $\gamma = 0.95$ for the MDP.

Finally, the *driver domain* was introduced in the work of Abbeel and Ng (2004), an instance of which is depicted in Fig. 7. In this environment, the agent corresponds to the driver of the blue car at the bottom, moving at a speed greater than all other cars. All



(a) Policy accur. (puddle-w.).

(b) Policy accur. (trap-w.).

(c) Policy accur. (driver).

(d) Value perf. (puddle-w.).

(e) Value perf. (trap-w.).

(f) Value perf. (driver).

Figure 8: Classification and value performance of GBS-IRL and IQBC in the three large domains. Solid lines correspond to GBS-IRL, and dotted lines correspond to IQBC. (b)-(c) Classification performance. (e)-(f) Value performance.

other cars move at constant speed and are scattered across the three central lanes. The goal of the agent is to drive as safely as possible—*i.e.,* avoid crashing into other cars, turning too suddenly and, if possible, driving in the shoulder lanes.

For the purposes of our tests, we represented the driver domain as an MDP with $|\mathcal{X}| = 16,875$ and $|\mathcal{A}| = 5$, where the five actions correspond to driving the car into each of the 5 lanes. Transitions are deterministic. The target reward function r^* penalizes the agent with a value of -10 for every crash, and with a value of -1 for driving in the shoulder lanes. Additionally, each lane change costs the agent a penalty of -0.1. As in the previous scenarios, we used a discount $\gamma = 0.95$ for the MDP.

As with the previous experiments, we conducted 200 independent learning trials for each of the three environments, and evaluated the performance of both GBS-IRL and IQBC. The results are depicted in Fig. 8.

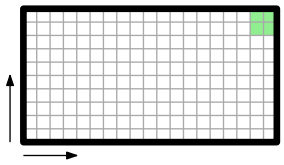


Figure 9: The grid-world used to illustrate the combined use of action and reward feedback.

We can observe that, as in previous scenarios, the performance of both methods is very similar. All scenarios feature a relatively small number of actions, which attenuates the negative dependence of IQBC on the number of actions observed in the previous experiments.

It is also interesting to observe that the trap-world domain seems to be harder to learn than the other two domains, in spite of the differences in dimension. For example, while the driver domain required only around 10 samples for GBS-IRL to single out the correct hypothesis, the trap-world required around 20 to attain a similar performance. This may be due to the fact that the trap-world domain features the sparsest reward. Since the other rewards in the hypothesis space were selected to be similarly sparse, it is possible that many would lead to similar policies in large parts of the state-space, thus hardening the identification of the correct hypothesis.

To conclude, is is still interesting to observe that, in spite of the dimension of the problems considered, both methods were effectively able to single out the correct hypothesis after only a few samples. In fact, the overall performance is superior to that observed in the medium-sized domains, which indicates that the domain structure present in these scenarios greatly contributes to disambiguate between hypothesis, given the expert demonstration.

4.2 Using Action and Reward Feedback

To conclude the empirical validation of our approach, we conduct a final set of experiments that aims at illustrating the applicability of our approach in the presence of both action and reward feedback.

One first experiment illustrates the integration of both reward and policy information in the Bayesian IRL setting described in Section 3.3. We consider the simple 19×10 grid-world depicted in Fig. 9, where the agent must navigate to the top-right corner of the environment. In this first experiment, we use random sampling, in which, at each time step t, the expert adds

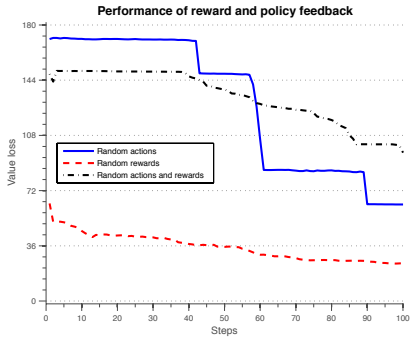
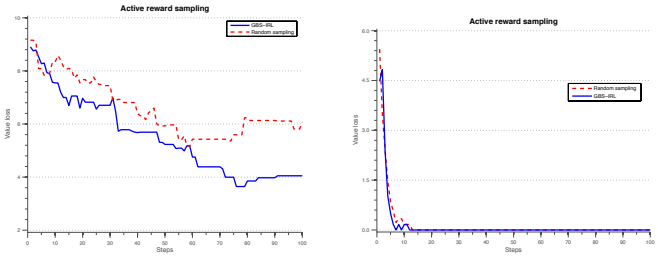


Figure 10: Bayesian IRL using reward and action feedback.



(a) (b)

Figure 11: Active IRL using reward feedback: sparse vs dense rewards.

one (randomly selected) sample to the demonstration $\mathcal{F}_t$, which can be of either form (x_t, a_t) or (x_t, u_t).

Figure 10 compares the performance of Bayesian IRL for demonstrations consisting of state-action pairs only, state-reward pairs only, and also demonstrations that include both state-action and state-reward pairs.

We first observe that all demonstration types enable the learner to slowly improve its performance in the target task. This indicates that all three sources of information (action, reward, and action+reward) give useful information to accurately identify the target task (or, equivalently, identify the target reward function).

Another important observation is that a direct comparison between the learning performance obtained with the different demonstration types may be misleading, since the ability of the agent to extract useful information from the reward samples greatly depends on the sparsity of the reward function. Except in those situations in which the reward is extremely informative, an action-based demonstration will generally be more informative.

In a second experiment, we analyze the performance of our active learning method when querying only reward information in the same grid-world environment. In particular, we analyze the dependence of the performance on the sparsity of the reward function, testing GBS-IRL in two distinct conditions. The first condition, depicted in Fig. 11(a), corresponds to a reward function r^* that is sparse, *i.e.,* such that $r^*(x) = 0$ for all states x except the goal states, where $r^*(x_{\text{goal}}) = 1$.

As discussed in Section 3.3, sparsity of rewards greatly impacts the learning performance of our Bayesian IRL approach. This phenomenon, however, is not exclusive to the active learning approach—in fact, as seen from Fig. 11(a), random sampling also exhibits a poor performance. It is still possible, nonetheless, to detect some advantage in using an active sampling approach.

In contrast, it is possible to design very informative rewards, by resorting to a technique proposed in the reinforcement learning literature under the designation of *reward shaping* (Ng et al, 1999). By considering a shaped version of that same reward, we obtain the learning performance depicted in Fig. 11(b). Note how, in the latter case, convergence is extremely fast even in the presence of random sampling.

We conclude by noting that, in the case of reward information, our setting is essentially equivalent to a standard reinforcement learning setting, for which efficient exploration techniques have been proposed and may provide fruitful avenues for future research.

5 Discussion

In this paper we introduce GBS-IRL, a novel active IRL algorithm that allows an agent to learn a task from a demonstration by an "expert". Using a generalization of binary search, our algorithm greedily queries the expert for demonstrations in highly informative states. As seen in Section 1.1, and following the designation of Dasgupta (2011), GBS-IRL is an *aggressive* active learning algorithm. Additionally, given our consideration of noisy samples, GBS-IRL is naturally designed to consider *non-separable data*. As pointed out by Dasgupta (2011), few aggressive active learning algorithms exist with provable complexity bounds for the non-separable case. GBS-IRL comes with such guarantees, summarized in Corollary 1: under suitable conditions and for any given $\delta > 0$, $\mathbb{P}\left[\mathbf{h}_t \neq \mathbf{h}^*\right] > 1 - \delta$, as long as

$$t \geq \frac{1}{\lambda} \log \frac{|\mathcal{H}|}{\delta},$$

where λ is a constant that does not depend on the dimension of the hypothesis space but only on the sample noise.

Additionally, as briefly remarked in Section 3.2, it is possible to use an adaptive sub-modularity argument to establish the near-optimality of GBS-IRL. In fact, given the target hypothesis, $\mathbf{h}^*$, consider the objective function

$$f(\mathcal{F}_t) = \mathbb{P}\left[h_t \neq h^* \mid \mathcal{F}_t\right] = 1 - p_t(h^*).$$

From Theorem 1 and its proof, it can be shown that f is *strongly adaptive monotone* and *adaptive sub modular* and use results of Golovin and Krause (2011) to provide a similar bound on sample complexity of GBS-IRL. To our knowledge, GBS-IRL is the first active IRL algorithm with provable sample complexity bounds. Additionally, as discussed in Section 3.2, our reduction of IRL to a standard (multi-class) classification problem implies that Algorithm 1 is not specialized in any particular way to IRL problems. In particular, our results are generally applicable in any multi-class classification problems verifying the corresponding assumptions.

Finally, our main contributions are focused in the simplest form of interaction, when the demonstration consist of examples of the right action to take in different situations. However, we also discuss how other forms of expert feedback (beyond policy information) may be integrated in a seamless manner in our GBS-IRL framework. In particular, we discussed how to combine both policy and reward information in our learning algorithm. Our approach thus provides an interesting bridge between reinforcement learning (or learning by trial and error) and imitation learning (or learning from demonstration). In particular, it brings to the forefront existing results on efficient exploration in reinforcement learning (Jaksch et al, 2010).

Additionally, the general Bayesian IRL framework used in this paper is also amenable to the integration of additional information sources. For example, the human agent may provide trajectory information, or indicate states that are frequently visited when following the optimal path. From the MDP parameters it is generally possible to associate a likelihood with such feedback, which can in turn be integrated in the Bayesian task estimation setting. However, extending the active learning approach to such sources of information is less straightforward and is left as an important avenue for future research.

A Proofs

In this appendix we collect the proofs of all statements throughout the paper.

A.1 Proof of Lemma 1

The method of proof is related to that of Nowak (2011). We want to show that either

- $W(p, [x]_i) \leq c^*$ for some $\mathcal{X}_i \in \Xi$ or, alternatively,
- There are two k-neighbor sets $\mathcal{X}_i, \mathcal{X}_j \in \Xi$ such that $W(p, [x]_i) > c^*$ and $W(p, [x]_j) > c^*$, while $A^*(p, [x]_i) \neq A^*(p, [x]_j)$.

We have that, for any $a \in \mathcal{A}$,

$$\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h}) \sum_{i=1}^{N} h([x]_i, a)\mu(\mathcal{X}_i) \leq \sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})c^* = c^*$$

The above expression can be written equivalently as

$$\mathbb{E}_\mu \left[\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_i, a)\right] \leq c^*. \tag{12}$$

Suppose that there is no $x \in \mathcal{X}$ such that $W(p, x) \leq c^*$. In other words, suppose that, for every $x \in \mathcal{X}$, $W(p, x) > c^*$. Then, for (12) to hold, there must be $\mathcal{X}_i, \mathcal{X}_j \in \Xi$ and $a \in \mathcal{A}$ such that

$$\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_i, a) > c^*$$
$$\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_j, a) < -c^*.$$

Since $(\mathcal{X}, \mathcal{H})$ is k-neighborly by assumption, there is a sequence $\{\mathcal{X}_{k_1}, \ldots, \mathcal{X}_{k_\ell}\}$ such that $\mathcal{X}_{k_1} = \mathcal{X}_i$, $\mathcal{X}_{k_\ell} = \mathcal{X}_j$, and every two sets $\mathcal{X}_{k_m}, \mathcal{X}_{k_{m+1}}$ are k-neighborly. Additionally, at some point in this sequence, the signal of $\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_i, a)$ must change. This implies that there are two k-neighboring sets $\mathcal{X}_{k_i}$ and $\mathcal{X}_{k_j}$ such that

$$\sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_{k_i}, a) > c^* \quad \sum_{\mathbf{h}\in\mathcal{H}} p(\mathbf{h})h([x]_{k_j}, a) < -c^*,$$

which implies that

$$A^*(p_t, [x]_{k_i}) \neq A^*(p_t, [x]_{k_j}),$$

and the proof is complete.

A.2 Proof of Theorem 1

Let C_t denote the amount of probability mass placed on incorrect hypothesis by p_t, i.e.,

$$C_t = \frac{1 - p_t(\mathbf{h}^*)}{p_t(\mathbf{h}^*)}.$$

The proof of Theorem 1 relies on the following fundamental lemma, whose proof can be found in Appendix A.3.

Lemma 2. *Under the conditions of Theorem 1, the process $\{C_t, t = 1, \ldots\}$ is a non-negative supermartingale with respect to the filtration $\{\mathcal{F}_t, t = 1, \ldots\}$. In other words,*

$$\mathbb{E}\left[C_{t+1} \mid \mathcal{F}_t\right] \leq C_t,$$

for all $t \geq 0$.

The proof now replicates the steps in the proof of Theorem 3 of Nowak (2011). In order to keep the paper as self-contained as possible, we repeat those steps here. We have that

$$\mathbb{P}\left[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\right] \leq \mathbb{P}\left[p_t(\mathbf{h}^*) < 1/2\right] = \mathbb{P}\left[C_t > 1\right] \leq \mathbb{E}\left[C_t\right],$$

where the last inequality follows from the Markov inequality. Explicit computations yield

$$\begin{aligned}
\mathbb{P}\left[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\right] &\leq \mathbb{E}\left[C_t\right] \\
&= \mathbb{E}\left[\frac{C_t}{C_{t-1}} C_{t-1}\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{C_t}{C_{t-1}} C_{t-1} \mid \mathcal{F}_{t-1}\right]\right] \\
&= \mathbb{E}\left[\mathbb{E}\left[\frac{C_t}{C_{t-1}} \mid \mathcal{F}_{t-1}\right] C_{t-1}\right] \\
&\leq \max_{\mathcal{F}_{t-1}} \mathbb{E}\left[\frac{C_t}{C_{t-1}} \mid \mathcal{F}_{t-1}\right] \mathbb{E}\left[C_{t-1}\right].
\end{aligned}$$

Finally, expanding the recursion,

$$\mathbb{P}\left[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\right] \leq C_0 \left(\max_{\tau=1,\ldots,t-1} \mathbb{E}\left[\frac{C_\tau}{C_{\tau-1}} \mid \mathcal{F}_{\tau-1}\right]\right)^t. \tag{13}$$

Since, from Lemma 2, $\mathbb{E}\left[\frac{C_t}{C_{t-1}} \mid \mathcal{F}_{t-1}\right] < 1$ for all t, the conclusion follows.

A.3 Proof of Lemma 2

The structure of the proof is similar to that of the proof of Lemma 2 of Nowak (2011). We start by explicitly writing the expression for the Bayesian update in (6). For all $a \in \mathcal{A}$, let

$$\begin{aligned}
\delta(a) &\triangleq \mathbb{P}_{p_t}\left[h(x_{t+1}, a) = 1\right] = \\
&= \tfrac{1}{2}\left(1 + \textstyle\sum_{\mathbf{h} \in \mathcal{H}} p_t(\mathbf{h}) h(x_{t+1}, a)\right),
\end{aligned} \tag{14}$$

and we abusively write δ_{t+1} to denote $\delta(a_{t+1})$. The quantity $\delta(a)$ corresponds to the fraction of probability mass concentrated on hypotheses prescribing action a as optimal in state x_{t+1}. The normalizing factor in the update (6) is given by

$$\begin{aligned}
&\sum_{\mathbf{h} \in \mathcal{H}} p_t(\mathbf{h}) \hat{\gamma}(x_{t+1})^{(1+h_{t+1})/2} \hat{\beta}(x_{t+1})^{(1-h_{t+1})/2} \\
&\quad = \sum_{\mathbf{h}: h_{t+1}=1} p_t(\mathbf{h}) \hat{\gamma}(x_{t+1}) + \sum_{\mathbf{h}: h_{t+1}=-1} p_t(\mathbf{h}) \hat{\beta}(x_{t+1}) \\
&\quad = \delta_{t+1} \hat{\gamma}(x_{t+1}) + (1 - \delta_{t+1}) \hat{\beta}(x_{t+1}).
\end{aligned}$$

We can now write the Bayesian update of $p_t(\mathbf{h})$ as

$$p_{t+1}(\mathbf{h}) = p_t(\mathbf{h}) \frac{\hat{\gamma}(x_{t+1})^{(1+h_{t+1})/2} \hat{\beta}(x_{t+1})^{(1-h_{t+1})/2}}{\delta_{t+1} \hat{\gamma}(x_{t+1}) + (1 - \delta_{t+1}) \hat{\beta}(x_{t+1})}. \tag{15}$$

Let

$$\eta(a) = \frac{\delta(a) \hat{\gamma}(x_{t+1}) + (1 - \delta(a)) \hat{\beta}(x_{t+1})}{\hat{\gamma}(x_{t+1})^{(1+h^*(x_{t+1},a))/2} \hat{\beta}(x_{t+1})^{(1-h^*(x_{t+1},a))/2}}, \tag{16}$$

where, as with δ, we abusively write η_{t+1} to denote $\eta(a_{t+1})$. Then, for $\mathbf{h}^*$, we can now write the update (15) simply as $p_{t+1}(\mathbf{h}^*) = p_t(\mathbf{h}^*)/\eta_{t+1}$, and

$$\frac{C_{t+1}}{C_t} = \frac{(1 - p_t(\mathbf{h}^*)/\eta_{t+1})\eta_{t+1}}{1 - p_t(\mathbf{h}^*)} = \frac{\eta_{t+1} - p_t(\mathbf{h}^*)}{1 - p_t(\mathbf{h}^*)}$$

The conclusion of the Lemma holds as long as $\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right] \leq 1$. Conditioning the expectation $\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right]$ on x_{t+1}, we have that

$$\begin{aligned}
&\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1}\right] \\
&\quad = \sum_{a \in \mathcal{A}} \eta(a) \mathbb{P}\left[a_{t+1} = a \mid \mathcal{F}_t, x_{t+1}\right] \\
&\quad = \sum_{a \in \mathcal{A}} \eta(a) \gamma^*(x_{t+1})^{(1+h^*(x_{t+1},a))/2} \beta^*(x_{t+1})^{(1-h^*(x_{t+1},a))/2}.
\end{aligned}$$

Let a^* denote the action in $\mathcal{A}$ such that $h^*(x_{t+1}, a^*) = 1$. This leads to

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1}\right] = \eta(a^*)\gamma^*(x_{t+1}) + \sum_{a \neq a^*} \eta(a)\beta^*(x_{t+1}). \tag{17}$$

For simplicity of notation, we temporarily drop the explicit dependence of β^*, $\hat{\beta}$, γ^* and $\hat{\gamma}$ on x_{t+1}. Explicit computations now yield

$$\begin{aligned}
&\eta(a^*)\gamma^*(x_{t+1}) + \sum_{a \neq a^*} \eta(a)\beta^*(x_{t+1}) \\
&\quad = [\delta(a^*)\hat{\gamma} + (1 - \delta(a^*))\hat{\beta}]\frac{\gamma^*}{\hat{\gamma}} + \\
&\quad + \sum_{a \neq a^*} [\delta(a)\hat{\gamma} + (1 - \delta(a))\hat{\beta}]\frac{\beta^*}{\hat{\beta}} \\
&\quad = \delta(a^*)\gamma^* + (1 - \delta(a^*))\frac{\hat{\beta}\gamma^*}{\hat{\gamma}} + \\
&\quad + \sum_{a \neq a^*} \left[\delta(a)\frac{\hat{\gamma}\beta^*}{\hat{\beta}} + (1 - \delta(a))\beta^*\right].
\end{aligned}$$

Since $\sum_{a\neq a^*}\delta(a) = 1-\delta(a^*)$,

$$\begin{aligned}
&\eta(a^*)\gamma^*(x_{t+1}) + \sum_{a\neq a^*}\eta(a)\beta^*(x_{t+1})\\
&\quad= (1-\delta(a^*))\left[\frac{\hat\beta\gamma^*}{\hat\gamma}+\frac{\hat\gamma\beta^*}{\hat\beta}\right]+\\
&\quad+\delta(a^*)\gamma^* + \sum_{a\neq a^*}(1-\delta(a))\beta^*\\
&\quad= (1-\delta(a^*))\left[\frac{\hat\beta\gamma^*}{\hat\gamma}+\frac{\hat\gamma\beta^*}{\hat\beta}\right]+\\
&\quad+\delta(a^*)\gamma^* + (|\mathcal{A}|-1)\beta^* - (1-\delta(a^*))\beta^*\\
&\quad= (1-\delta(a^*))\left[\frac{\hat\beta\gamma^*}{\hat\gamma}+\frac{\hat\gamma\beta^*}{\hat\beta}\right]+\\
&\quad+\delta(a^*)(\gamma^*+\beta^*) + 1 - \gamma^* - \beta^*,
\end{aligned}$$

where we have used the fact that $(|A|-1)\beta^*+\gamma^* = 1$. Finally, we have

$$\begin{aligned}
&\eta(a^*)\gamma^*(x_{t+1}) + \sum_{a\neq a^*}\eta(a)\beta^*(x_{t+1})\\
&= (1-\delta(a^*))\left[\frac{\hat\beta\gamma^*}{\hat\gamma}+\frac{\hat\gamma\beta^*}{\hat\beta}-\gamma^*-\beta^*\right]+1\\
&= 1-(1-\delta(a^*))\left[\gamma^*+\beta^*-\frac{\hat\beta\gamma^*}{\hat\gamma}-\frac{\hat\gamma\beta^*}{\hat\beta}\right]\\
&= 1-(1-\delta(a^*))\left[\gamma^*\frac{\hat\gamma-\hat\beta}{\hat\gamma}+\beta^*\frac{\hat\beta-\hat\gamma}{\hat\beta}\right].
\end{aligned}$$

Letting $\rho = 1-(|\mathcal{A}|-1)\alpha$, we have that $\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1}\right] < 1$ as long as

$$\gamma^*(x)\frac{\hat\gamma(x)-\hat\beta(x)}{\hat\gamma(x)} + \beta^*(x)\frac{\hat\beta(x)-\hat\gamma(x)}{\hat\beta(x)} > 0$$

for all $x\in\mathcal{X}$. Since, for all $x\in\mathcal{X}$, $\beta^*(x) < \alpha$ and $\gamma^*(x) > \rho$, we have

$$\begin{aligned}
&\gamma^*(x)\frac{\hat\gamma(x)-\hat\beta(x)}{\hat\gamma(x)} + \beta^*(x)\frac{\hat\beta(x)-\hat\gamma(x)}{\hat\beta(x)} =\\
&= (\gamma^*(x)-\beta^*(x))\left[\frac{\gamma^*(x)}{\hat\gamma(x)}-\frac{\beta^*(x)}{\hat\beta(x)}\right]\\
&> (\rho-\alpha)\left[\frac{\rho}{\hat\gamma(x)}-\frac{\alpha}{\hat\beta(x)}\right] \geq 0.
\end{aligned}$$

where the inequality is strict if $\hat\beta(x) > \alpha$ for all $x\in\mathcal{X}$.

A.4 Proof of Theorem 2

To prove Theorem 2, we depart from (13):

$$\mathbb{P}\big[\hat{\mathbf{h}}_t \neq \mathbf{h}^*\big] \leq C_0\left(\max_{\tau=1,\ldots,t-1}\mathbb{E}\left[\frac{C_\tau}{C_{\tau-1}} \mid \mathcal{F}_{\tau-1}\right]\right)^t.$$

Letting

$$\lambda_t = \max_{\tau=1,\ldots,t-1}\mathbb{E}\left[\frac{C_\tau}{C_{\tau-1}} \mid \mathcal{F}_{\tau-1}\right],$$

the desired result can be obtained by bounding the sequence $\{\lambda_t, t=0,\ldots\}$ by some value $\lambda<1$. To show that such λ exists, we consider separately the two possible queries in Algorithm 1.

Let then $c_t = \min_{i=1,\ldots,N} W(p_t,[x]_i)$, and suppose that there are no 1-neighbor sets $\mathcal{X}_i$ and $\mathcal{X}_j$ such that

$$W(p_t,[x]_i) > c_t, \qquad W(p_t,[x]_j) > c_t, \tag{18}$$
$$A^*(p_t,[x]_i) \neq A^*(p_t,[x]_j). \tag{19}$$

Then, from Algorithm 1, the queried state x_{t+1} will be such that

$$x_{t+1} \in \operatorname*{argmin}_i W(p_t,[x]_i).$$

Since, from the definition of c^*, $c_t < c^*$, it follows that $\delta(a) \leq \frac{1+c^*}{2}$ for all $a\in\mathcal{A}$, where $\delta(a)$ is defined in (14). Then, from the proof of Lemma 2,

$$\begin{aligned}
\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1}\right] &\leq 1-\varepsilon(1-\delta(a^*))\\
&\leq 1-\varepsilon\frac{1-c^*}{2},
\end{aligned}$$

where a^* denotes the action in $\mathcal{A}$ such that $h^*(x,a^*) = 1$.

Consider now the case where there are 1-neighboring sets $\mathcal{X}_i$ and $\mathcal{X}_j$ such that (18) holds. In this case, according to Algorithm 1, x_{t+1} is selected randomly as either $[x]_i$ or $[x]_j$ with probability 1/2. Moreover, since $\mathcal{X}_i$ and $\mathcal{X}_j$ are 1-neighbors, there is a single hypothesis, say $\mathbf{h}_0$, that prescribes different optimal actions in $\mathcal{X}_i$ and $\mathcal{X}_j$. Let a_i^* denote the optimal action at $[x]_i$, and a_j^* the optimal action at $[x]_j$, as prescribed by $\mathbf{h}^*$. Three situations are possible:

Situation 1. $A^*(p_t,[x]_i) \neq a_i^*$ and $A^*(p_t,[x]_j) = a_j^*$, or $A^*(p_t,[x]_i) = a_i^*$ and $A^*(p_t,[x]_j) \neq a_j^*$.

Situation 2. $A^*(p_t,[x]_i) \neq a_i^*$ and $A^*(p_t,[x]_j) \neq a_j^*$;

Situation 3. $A^*(p_t,[x]_i) = a_i^*$ and $A^*(p_t,[x]_j) = a_j^*$;

We consider Situation 1 first. From the proof of Lemma 2,

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1} \in \{[x]_i, [x]_j\}\right] \leq 1 - \frac{\varepsilon}{2}\left[1 - \frac{1}{2}\sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})\big(h([x]_i, a_i^*) + h([x]_j, a_j^*)\big)\right],$$

where we explicitly replaced the definition of $\delta(a)$. If $A^*(p_t, [x]_i) = a_i^*$ and $A^*(p_t, [x]_j) \neq a_j^*$ (the alternative is treated similarly), we have that

$$\sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})h([x]_i, a_i^*) \leq 1 \quad \text{and} \quad \sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})h([x]_i, a_i^*) \leq 0,$$

yielding

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1} \in \{[x]_i, [x]_j\}\right] \leq 1 - \frac{\varepsilon}{4}.$$

Considering Situation 2, we again have

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1} \in \{[x]_i, [x]_j\}\right] \leq 1 - \frac{\varepsilon}{2}\left[1 - \frac{1}{2}\sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})\big(h([x]_i, a_i^*) + h([x]_j, a_j^*)\big)\right],$$

where, now,

$$\sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})h([x]_i, a_i^*) \leq 0 \quad \text{and} \quad \sum_{\mathbf{h}\in\mathcal{H}} p_t(\mathbf{h})h([x]_i, a_i^*) \leq 0.$$

This immediately implies

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1} \in \{[x]_i, [x]_j\}\right] \leq 1 - \frac{\varepsilon}{2}.$$

Finally, concerning Situation 3, $\mathbf{h}_0 = \mathbf{h}^*$. Since $\mathcal{X}_i$ and $\mathcal{X}_j$ are 1-neighbors, $h([x]_i, a_i^*) = h([x]_j, a_i^*)$ for all hypothesis other than $\mathbf{h}^*$. Equivalently, $h([x]_i, a_i^*) = -h([x]_j, a_j^*)$ for all hypothesis other than $\mathbf{h}^*$. This implies that

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t, x_{t+1} \in \{[x]_i, [x]_j\}\right] \leq 1 - \frac{\varepsilon}{2}(1 - p_t(\mathbf{h}^*)).$$

Putting everything together,

$$\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right] \leq \max\left\{1 - \frac{\varepsilon}{4}, 1 - \frac{\varepsilon}{2}(1 - p_t(\mathbf{h}^*)), 1 - \frac{\varepsilon}{2}(1 - c^*)\right\}$$

and

$$\mathbb{E}\left[\frac{C_\tau}{C_{\tau-1}} \mid \mathcal{F}_{\tau-1}\right] \leq \frac{\mathbb{E}\left[\eta_{t+1} \mid \mathcal{F}_t\right] - p_t(\mathbf{h}^*)}{1 - p_t(\mathbf{h}^*)} \leq 1 - \min\left\{\frac{\varepsilon}{4}, \frac{\varepsilon}{2}(1 - c^*)\right\}.$$

The proof is complete. □

A.5 Proof of Theorem 3

Let $\varepsilon = 1 - \hat{c}$ and

$$C_t = \frac{\varepsilon - p_t(\mathbf{h}^*)}{p_t(\mathbf{h}^*)}.$$

Let a denote an arbitrary action in $\mathcal{A}_{\hat{c}}(p_t, [x]_i)$, for some $[x]_i, i = 1, \ldots, N$. Then

$$\begin{aligned}
&\mathbb{P}\left[h^*([x]_i, a) = -1\right] \\
&= \mathbb{P}\left[\sum_{\mathbf{h}\neq\mathbf{h}^*} p_t(\mathbf{h})h([x]_i, a) > \hat{c} + p_t(\mathbf{h})\right] \\
&\leq \mathbb{P}\left[\sum_{\mathbf{h}\neq\mathbf{h}^*} p_t(\mathbf{h}) > \hat{c} + p_t(\mathbf{h})\right] \\
&= \mathbb{P}\left[1 - p_t(\mathbf{h}^*) > \hat{c} + p_t(\mathbf{h})\right] \\
&= \mathbb{P}\left[C_t > 1\right] \\
&\leq \mathbb{E}\left[C_t\right],
\end{aligned}$$

where, again, the last inequality follows from the Markov inequality. We can now replicate the steps in the proof of Theorem 1 in Appendix A.2 to establish the desired result, for which we need only to prove that

$$\mathbb{E}\left[C_{t+1} \mid \mathcal{F}_t\right] \leq C_t.$$

From Lemma 2, the result follows. □

Acknowledgements

This work was partially supported by the Portuguese Fundação para a Ciência e a Tecnologia (INESC-ID multiannual funding) under project PEst-OE/EEI/LA0021/2011. Manuel Lopes is with the Flowers Team, a joint INRIA ENSTA-Paristech lab.

References

Abbeel P, Ng A (2004) Apprenticeship learning via inverse reinforcement learning. In: Proc. 21st Int. Conf. Machine Learning, pp 1–8

Argall B, Chernova S, Veloso M (2009) A survey of robot learning from demonstration. Robotics and Autonomous Systems 57(5):469–483

Babes M, Marivate V, Littman M, Subramanian K (2011) Apprenticeship learning about multiple intentions. In: Proc. 28th Int. Conf. Machine Learning, pp 897–904

Barto A, Rosenstein M (2004) Supervised actor-critic reinforcement learning. In: Si J, Barto A, Powell W, Wunsch D (eds) Handbook of Learning and Approximate Dynamic Programming, Wiley-IEEE Press, chap 14, pp 359–380

Boyan J, Moore A (1995) Generalization in reinforcement learning: Safely approximating the value function. In: Adv. Neural Information Proc. Systems, vol 7, pp 369–376

Breazeal C, Brooks A, Gray J, Hoffman G, Lieberman J, Lee H, Thomaz A, Mulanda D (2004) Tutelage and collaboration for humanoid robots. Int J Humanoid Robotics 1(2)

Cakmak M, Thomaz A (2010) Optimality of human teachers for robot learners. In: Proc. 2010 IEEE Int. Conf. Development and Learning, pp 64–69

Chernova S, Veloso M (2009) Interactive policy learning through confidence-based autonomy. J Artificial Intelligence Res 34:1–25

Cohn R, Durfee E, Singh S (2011) Comparing action-query strategies in semi-autonomous agents. In: Proc. 25th AAAI Conf. Artificial Intelligence, pp 1102–1107

Dasgupta S (2005) Analysis of a greedy active learning strategy. In: Adv. Neural Information Proc. Systems, vol 17, pp 337–344

Dasgupta S (2011) Two faces of active learning. J Theoretical Computer Science 412(19):1767–1781

Golovin D, Krause A (2011) Adaptive submodularity: Theory and applications in active learning and stochastic optimization. J Artificial Intelligence Res 42:427–486

Grollman D, Jenkins O (2007) Dogged learning for robots. In: Proc. 2007 IEEE Int. Conf. Robotics and Automation, pp 2483–2488

Jaksch T, Ortner R, Auer P (2010) Near-optimal regret bounds for reinforcement learning. J Machine Learning Res 11:1563–1600

Judah K, Fern A, Dietterich T (2011) Active imitation learning via state queries. In: Proc. ICML Workshop on Combining Learning Strategies to Reduce Label Cost

Judah K, Fern A, Dietterich T (2012) Active imitation learning via reduction to I.I.D. active learning. In: Proc. 28th Conf. Uncertainty in Artificial Intelligence, pp 428–437

Knox W, Stone P (2009) Interactively shaping agents via human reinforcement. In: Proc. 5th Int. Conf. Knowledge Capture, pp 9–16

Knox W, Stone P (2010) Combining manual feedback with subsequent MDP reward signals for reinforcement learning. In: Proc. 9th Int. Conf. Autonomous Agents and Multiagent Systems, pp 5–12

Knox W, Stone P (2011) Augmenting reinforcement learning with human feedback. In: IJCAI Workshop on Agents Learning Interactively from Human Teachers

Lopes M, Melo F, Kenward B, Santos-Victor J (2009a) A computational model of social-learning mechanisms. Adaptive Behavior 467(17)

Lopes M, Melo F, Montesano L (2009b) Active learning for reward estimation in inverse reinforcement learning. In: Proc. Eur. Conf. Machine Learning and Princ. Practice of Knowledge Disc. Databases, pp 31–46

Lopes M, Melo F, Montesano L, Santos-Victor J (2010) Abstraction levels for robotic imitation: Overview and computational approaches. In: Sigaud O, Peters J (eds) From Motor to Interaction Learning in Robots, Springer, pp 313–355

Melo F, Lopes M (2010) Learning from demonstration using MDP induced metrics. In: Proc. European Conf. Machine Learning and Practice of Knowledge Discovery in Databases, pp 385–401

Melo F, Lopes M, Santos-Victor J, Ribeiro M (2007) A unified framework for imitation-like behaviours. In: Proc. 4th Int. Symp. Imitation in Animals and Artifacts, pp 241–250

Melo F, Lopes M, Ferreira R (2010) Analysis of inverse reinforcement learning with perturbed demonstrations. In: Proc. 19th European Conf. Artificial Intelligence, pp 349–354

Neu G, Szepesvari C (2007) Apprenticeship learning using inverse reinforcement learning and gradient methods. In: Proc. 23rd Conf. Uncertainty in Artificial Intelligence, pp 295–302

Ng A, Russel S (2000) Algorithms for inverse reinforcement learning. In: Proc. 17th Int. Conf. Machine Learning, pp 663–670

Ng A, Harada D, Russell S (1999) Policy invariance under reward transformations: Theory and application to reward shaping. In: Proc. 16th Int. Conf. Machine Learning, pp 278–294

Nowak R (2011) The geometry of generalized binary search. IEEE Trans Information Theory 57(12):7893–7906

Price B, Boutilier C (1999) Implicit imitation in multiagent reinforcement learning. In: Proc. 16th Int. Conf. Machine Learning, pp 325–334

Price B, Boutilier C (2003) Accelerating reinforcement learning through implicit imitation. J Artificial Intelligence Res 19:569–629

Ramachandran D, Amir E (2007) Bayesian inverse reinforcement learning. In: Proc. 20th Int. Joint Conf. Artificial Intelligence, pp 2586–2591

Regan K, Boutilier C (2011) Robust online optimization of reward-uncertain MDPs. In: Proc. 22nd Int. Joint Conf. Artificial Intelligence, pp 2165–2171

Ross S, Bagnell J (2010) Efficient reductions for imitation learning. In: 661-668 (ed) Proc. 13th Int. Conf. Artificial Intelligence and Statistics

Ross S, Gordon G, Bagnell J (2011) Reduction of imitation learning and structured prediction to no-regret online learning. In: Proc. 14th Int. Conf. Artificial Intelligence and Statistics, pp 627–635

Schaal S (1999) Is imitation learning the route to humanoid robots? Trends in Cognitive Sciences 3(6):233–242

Settles B (2009) Active learning literature survey. Comp. Sciences Techn. Rep. 1648, Univ. Wisconsin-Madison

Sutton R, Barto A (1998) Reinforcement Learning: An Introduction. MIT Press

Syed U, Schapire R (2008) A game-theoretic approach to apprenticeship learning. In: Adv. Neural Information Proc. Systems, vol 20, pp 1449–1456

Syed U, Schapire R, Bowling M (2008) Apprenticeship learning using linear programming. In: Proc. 25th Int. Conf. Machine Learning, pp 1032–1039

Thomaz A, Breazeal C (2008) Teachable robots: Understanding human teaching behavior to build more effective robot learners. Artificial Intelligence 172:716–737

Ziebart B, Maas A, Bagnell J, Dey A (2008) Maximum entropy inverse reinforcement learning. In: Proc. 23rd AAAI Conf. Artificial Intelligence., pp 1433–1438